UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Bachelor's degree in Industrial Electronics and Automatic Control Engineering

Design, construction, programming and monitoring of a testbench for brushless motors

Eduard Mainou Gobierno

Director: Antonio Miguel López Martínez

June 2018

Abstract

This document explains how to develop successfully a testbench for brushless motors connected wireless to a PC to analyze the data.

This dissertation involves all the development and implementation of all the parts of the final prototype, including workarounds and redesigns. It also includes all the technical information to reproduce such prototype and all the software and source code necessary for the tool.

Resum

Aquest document explica com desenvolupar satisfactoriament una bancada de proves per a motors de tipo *brushless* connectada sense fils a un PC per al analisi de dades.

Aquesta memoria inclou tot el desenvolupament i la implementació de totes les parts del prototip final, incloent tots els *workarounds* i redissenys. També conté tota la informació tecnica per reproduir el montatge del prototip i tot el codi font del software desenvolupat per a la eina de visualització.

Resumen

Este documento explica cómo desarrollar satisfactoriamente una bancada de pruebas para motores de tipo *brushless* conectada inalámbricamente a un PC para el análisis de datos.

Esta memoria incluye todo el desarrollo y la implementación de todas las partes del prototipo final, incluyendo todos los *workarounds* y rediseños. También contiene toda la información técnica para reproducir el montaje del prototipo y todo el código fuente del software desarrollado para la herramienta de visualización.

Acknowledgements

First of all, I want to say thanks to Venturi Unmanned Technologies SL. for trusting me with this project and helping with all the cost and showing unconditional support through all the months of work.

I want to acknowledge the dedication that my tutor, Antonio, has put into this project even with the tight schedule that he had during the same months.

The incredible support I received from my superiors and colleagues from FICOSA, which helped me through all the doubts I had, and covered my shifts and helped out with my tasks during the last weeks, where the amount of work I was carrying was inhuman.

Special thanks to my family and my friends, who stood by my side with anything that came out.

And to Ivan, my best friend, who carried my stress with him and was there whenever I needed to disconnect.

Table of Contents

Table of Figures

List of Tables

List of Acronyms

Acronym	Meaning
VUT	Venturi Unmanned Technologies
ADC	Analogic Digital Converter
USB	Universal Serial Bus
GUI	Graphic User Interface
UAV	Unmanned Aerial Vehicle
CTO	Chief Technical Operations
PC	Personal Computer
OA	Operational Amplifier
CSV	Comma Separated Values
COM PORT	Communication Port
LCD	Liquid Crystal Display

Introduction

Objectives

The main objective of this project was to develop a testbench that could fulfill all the requirements of VUT and create the optimal solution to their main problem. Renting a wind tunnel is uneconomical for a small company, thus they need a solution.

The easier way to simulate a wind tunnel is to attach a testbench to a car and drive it at a certain speed in one direction. Seeing that not even in the state of the art of testbenches existed one model that could storage multiple variable data, the idea of gathering the data in the testbench and send it using XBee[1] protocol for later storage and analysis came out.

Hence, these were the main requirements that the testbench should have achieved:

- Metrology capabilities
- Wireless communication
- Low cost
- Accuracy
- Upgradable design

Therefore, this project was developed simultaneously with their main product development, in order not to waste any resources and once their technical requirements on which magnitudes their test will require all the components were specified.

In order to complete the design and implement all the requirements the project had 3 phases. The first one was the design of the layout and physical structure of the product. The second was the development of the software and code that would allow the data to get collected in the testbench and send it wireless to a remote PC. Finally, the last stage was the implementation of the hardware and sensors to collect the data. Obtaining the hardware in the last stage of design reduced the cost of the project since less workarounds and test had to be done. All the functionalities could be tested with dummy data generated by a script or voltage dividers.

Purpose

This Bachelor's Final Degree Project was an assignment from the startup Venturi Unmanned Technologies.

The main goal was to develop a full functional testbench to correctly gather all the data from different tests performed on a brushless motor and send the data wireless to a remote system.

The principal constraints were time and budget, as this project had to be as affordable as possible and not halt the company development.

In order to develop their prototype of self-driving UAV, they needed a bench to test the different brushless motors candidates for their project.

Scope

The scope of this project was as ambitious as the company's vision of their comercial scope.

VUT is a start-up that is developing a watchkeeper UAV. That particular UAV requires 2 different sizes and types of brushless motors to be taken into account. Those types of brushless motors define the scope of the projects. The benchmark will be optimized to those motors. It is possible to test other types of motors, but the accuracy and ideal performance is not guaranteed

Schedule

Project Planner

DEADLINE: 40

Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
State of the art research	1	5	1	6	100%
Kick Off meeting with VUT	3	4	3	4	100%
Model Design	6	4	6	3	100%
Software Design	6	8	7	6	100%
Hardware Design	12	5	12	5	100%
Model Implementation	17	5	17	5	100%
Software Implementation	17	14	18	15	100%
Hardware Implementation	17	4	17	5	100%
Prototype Test #1	22	2	22	2	100%
Prototype Test #2	31	2	31	2	100%
Evaluation	32	3	32	3	100%
Documentation	20	20	21	19	100%

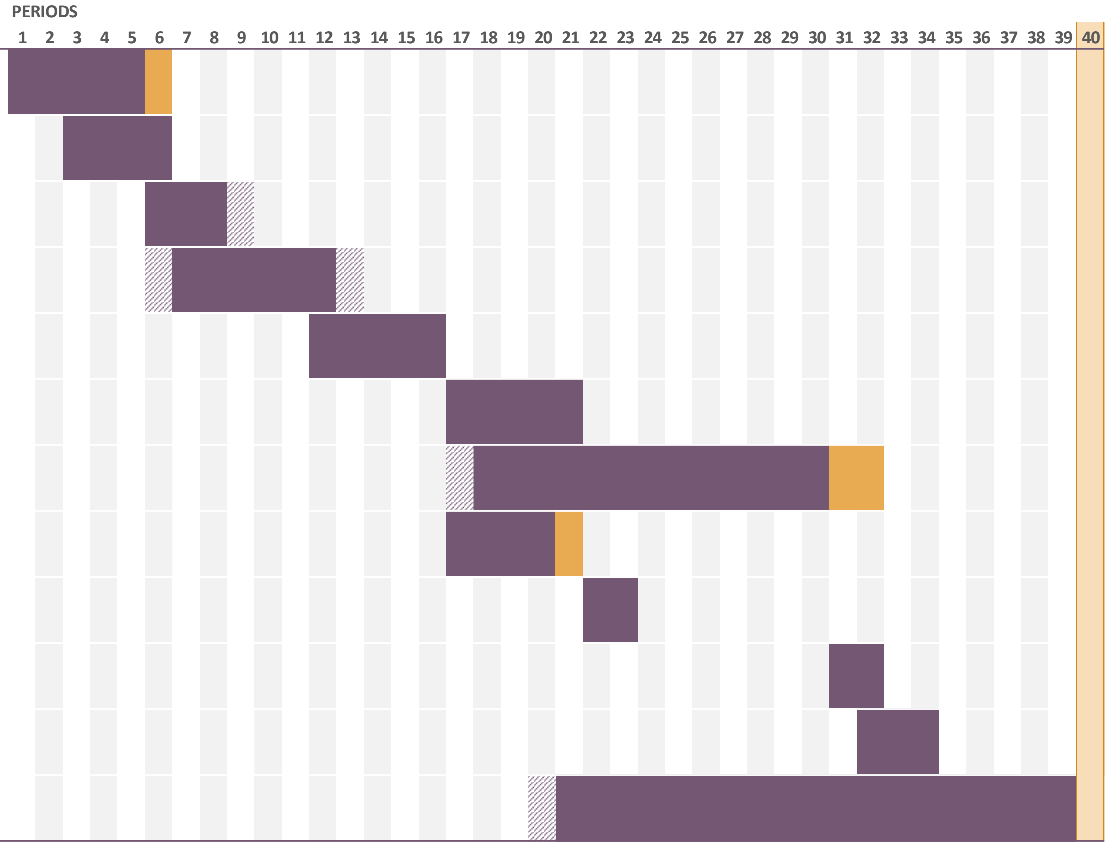


Figure 1 Gantt graph representing the time schedule of the project

Requirements

Stating the requirements for the asset was fundamental, as this was a custom project made for a specific purpose fulfilling the technical requirements was key for its usability.

VUT stated that in order to test their brushless motors correctly, the sensors on the bench should at least fulfill the limits and resolution stated the Table 1. They also listed the preference of priority in case that the first prototype could not include all of them.

Preference	Variable	Min	Max	Resolution
1	Thrust	0 kg	25 kg	0.1 kg
2	Tension	0 V	80 V	0.01 V
3	Current	0 A	200 A	0.01 A

Table 1 Requirements

Knowing all the technical requirements allowed the process to move onto selecting the sensors needed to measure the variables listed with the adequate resolution.

VUT did not list any requirement for the sample time of those variables so it was set to be the minimum in order to get the best performance, but not having that as a requirement granted so much flexibility in choosing the components for the communication.

As it was not a requirement but both the CTO of VUT and the director of this project had worked with XBee modules, those were the choice to maintain the communication between the sensors and the Arduino and the PC used to analyze the data.

Being all the specifications sorted out the process of design begins.

Development
Background

Doing a research in the field you are going to work on is a must in engineering fields.

The state of the art changes almost every month, hence your design shall start from the state of the art status and try to improve from there.

Concept
In this case, the start of the art of testbenches for brushless motors was in the lines of what RCBenchmark[2] have. They produce high quality and accuracy Thrust stands that can display the power consumption. The Series 1580 comes with a USB interface and software for automated control, data logging, and acquisition of mechanical efficiency.

Figure 2 Thrust stand from the brand RCBenchmark

Another company that produces testbenches is TURNIGY[3] power systems, their thrust stand and power analyser v2 as seen in Fig. 3 does include a small display to show the data, but this one does not even integrate a way to record and storage the data that reads.

Figure 3 Thrust stand from the brand TURNIGY

Nevertheless, none commercialized of these thrust stands or testbenches had any wireless data transmittance. This particular capacity was the one that VUT required the most, since they had to recreate a wind tunnel, implementing the custom testbench onto the top of a car.

Hardware

Having sort out all the State of the art embedded systems that can be bought on the market, a deeper investigation regarding the components of what the designed system should have was required.

Has explained before, since this was a low-cost project, we focused on open source components and software.

Nowadays there are so many Arduino and RasperryPi compatible components, but those cost extra. To avoid extra cost on components it was decided that as long as possible all the sensors would be made with basic electronic components such as Operational Amplifiers and resistors. This required some background and research on how to build the required sensors.

For the conversion from analog data to digital data, getting dedicated ADC with high specifications would rise the cost, hence VUT compromised with the integrated ADCs on Arduino's UNO board.

All of the information regarding the Arduino[4] UNO board with the ATmega328P microcontroller was in the official Arduino website.

All of the information regarding XBee modules was also free on their website as well the software to configure them. This made all the research and the practice that was needed to use these modules so much easier.

Software

Regarding on which language to use for the data acquisition on the PC, after doing some study, the language programmers use the most for the software that this project needed was Python[5]. Having all the libraries free as well, this keep the project on track for its budget and open source structure.

Once all the basic from the language was understood, it was needed to choose GUI for the users to interact with the software.

There are so many GUI listed on the Python Website as easy to use and intuitive but the one that seemed the faster to learn was Tkinter[6], which also its included with the Python package so no additional download is required.

In addition, when a software for simulate the data send by the XBee over a COMPort, a program named Com0Com[7], was used to simulate a virtual port.

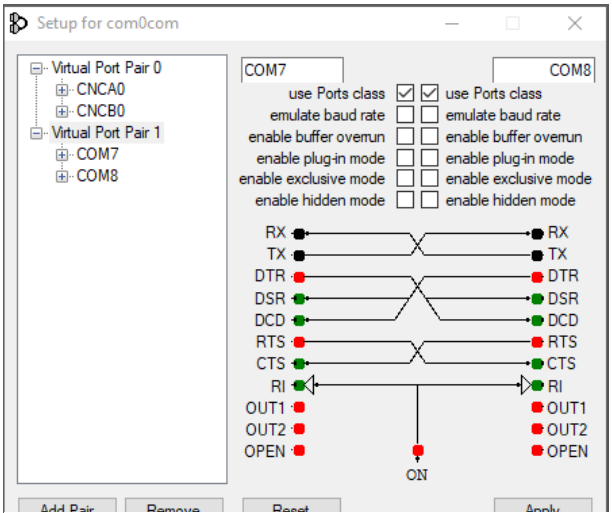


Figure 4 Setup window for the program com0com

Design

In this section, it is explained the design loop for the 3 main parts of the project, the physical model, the hardware of project as well as its software.

For more detailed and technical specifications, see implementation section.

Model design

The main idea was to design a structure that could fit on the top of a car to do the test that VUT required.

Solidworks[8] was used since it did not require to have the dimensions specified, it only had to be a sketch of how the final structure should look like, since it was not specified on which car would be placed.

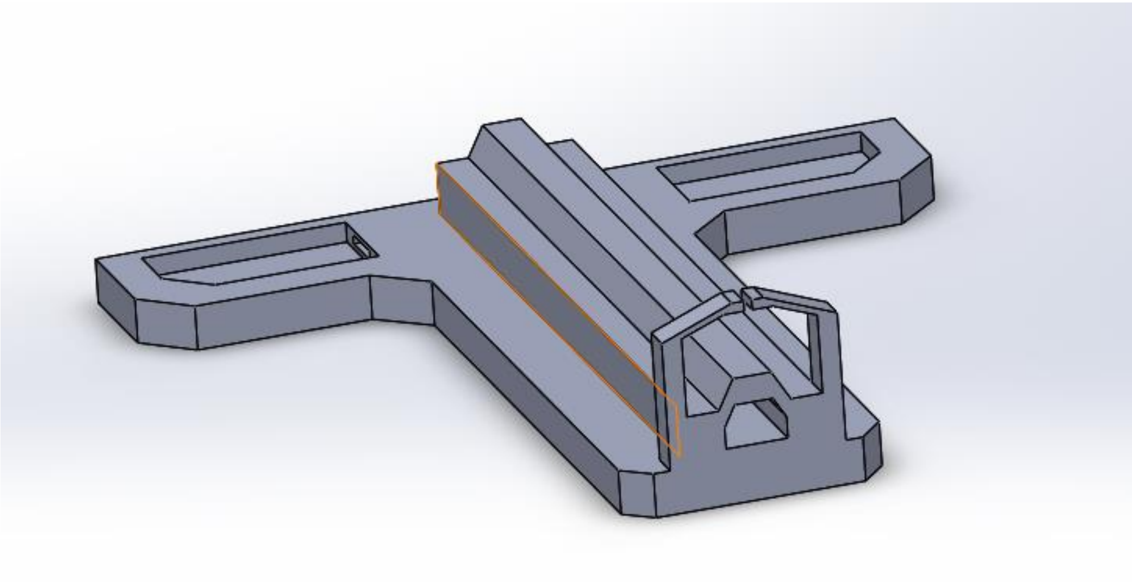


Figure 5 Design of the model with Solidworks 2017

The T shape was developed in a way that could create enough space for the hardware to be placed far away from the sensors and the motor. In addition, the hardware like the Arduino and the XBee could be placed in the areas on the side of the main line, adding more aesthetic look and keeping all the valuable pieces secured.

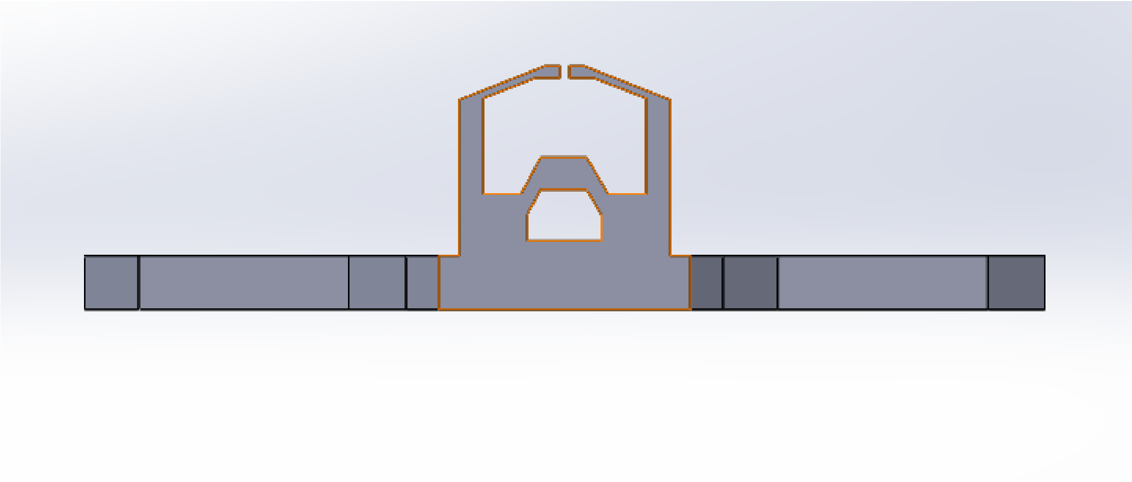


Figure 6 Front view of the model

In the fig. 6 it is represented the part where the motor would be placed, with the minimum amount of material around it, so it would not affect the airflow.

Also, we can see the hole inside the structure that would allow all the cables to go from the sensors near the motor to the back of the structure.

In the fig. 7, the aerial view allows to see the compartment where initially the expensive hardware would be, away from any interference and easy access to any upgrade.

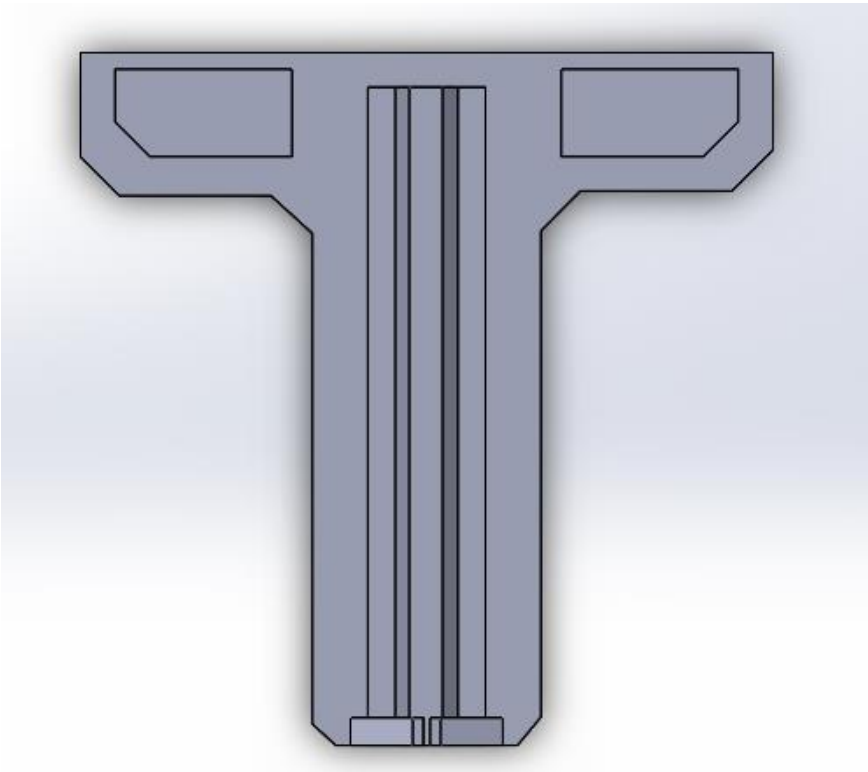


Figure 7 Aerial view of the model

Hardware design

All of the hardware design revolved around the XBee modules, hence in order to simply the design as much as possible to make it upgradable for people with limited electronic knowledge, an Arduino board, a XBee Explorer and breadboards level electronic components were used.

Figure 8 Digi International XBee S1 Pro

Figure 9 Arduino UNO Board

Figure 10 Sparkfun Explorer USB

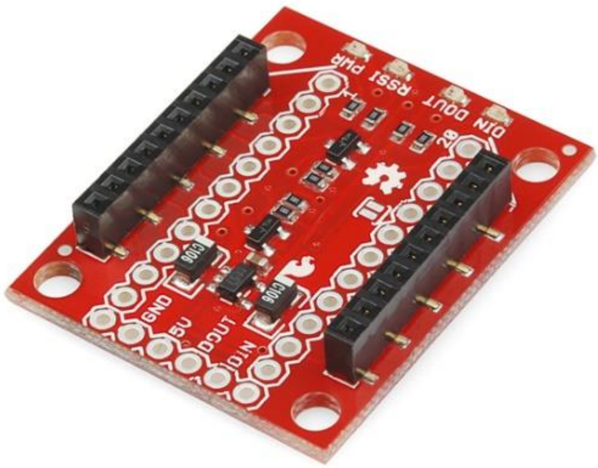


Figure 11 Sparkfun Breadboard Adapter

The Arduino board was an elemental choice, since it worked as an ADC and a 5 V power supply besides being a microcontroller that allowed us to manage the data of the ADC's and use the serial output to send that data over to the XBee.

The first layer of the hardware design were the sensors that would capture the magnitudes to measure.

At the beginning, using Instrumental amplifiers were considered, but two main problems were found, we did not have negative voltage nor enough differential voltage in order not to saturate an OA when we tried to lower the max input of 80 V to 5 V.

These two drawbacks forced the design to adapt and use a resistor divider with a buffer amplifier before the Arduino's ADC port. This was the main workaround of the hardware design. High magnitude resistors are needed in order that the drawback in current is less than 0.01% of what is being measured.

$$Vout = Vin * \frac{R_2}{R_1 + R_2}$$

$$5\ V = 80\ V * \frac{0.4\ M\Omega}{6\ M\Omega + 0.4\ M\Omega}$$

Choosing those exact reistors then the current drawback is:

$$0.0000125\ A = \frac{80\ V}{6\ M\Omega + 0.4\ M\Omega}$$

$$0.0000125\ \% = \frac{0.0000125\ A * 100}{100\ A}$$

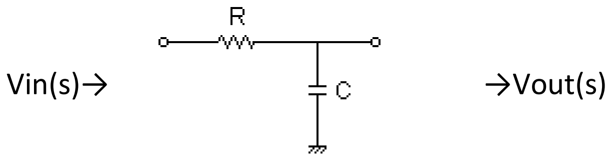
The current deviated is 0.0000125 % of the measure. This justifies the preference on high value resistors. In addition to defend the workaround as a possible solution.

A ACS758[9] was used to measure the current of the system. This sensor included its own downgrade to 0-5 V, which fitted perfect with the rest of low voltage system

To measure the thrust a strain gauge with an instrumentation amplifier such as a INA125[10] would be used and its output would go directly connected to the ADC.

Low-pass filters were calculated to prevent noise to invalidate the data send to the XBee. Any frequency higher than 10 Hz will be attenuated.

$$fc = \frac{1}{2\pi RC}$$



R:160 Ω

C: 100 µF

fc: 10 Hz

The main concern was not to let any of the high current of this layer trespass to the second layer that would be the Arduino and XBee boards, very sensitive devices to high currents and voltages.

The second layer of the hardware was the connections between the Arduino board and the XBee transmitter. An adapter to breadboard pins was used to connect both of them. Making it easier for the user to modify any component for malfunction or to change the resistors from the voltage divider in order to calibrate the system for a different range of voltage. This was necessary since not all the motors needed to be tested were powered by the same input, so in order not to lose precision on the voltmeter, the capacity to readjust the voltage divider was introduced in the design loop.

Figure 12 Breardboard

Figure 13Jumpers for breadboards

Finally all the low voltage system (5V) would be powered by a simple AA batteries pack, while all the sensors would be connected to the main source of power for the brushless motor which can reach up to 80 V and a 100 A. To prevent any overcurrent or overvoltage to the low voltage system a buffer between the voltage divider and the ADC of the arduino would be placed.

Figure 14 AA batteries adapter for Arduino

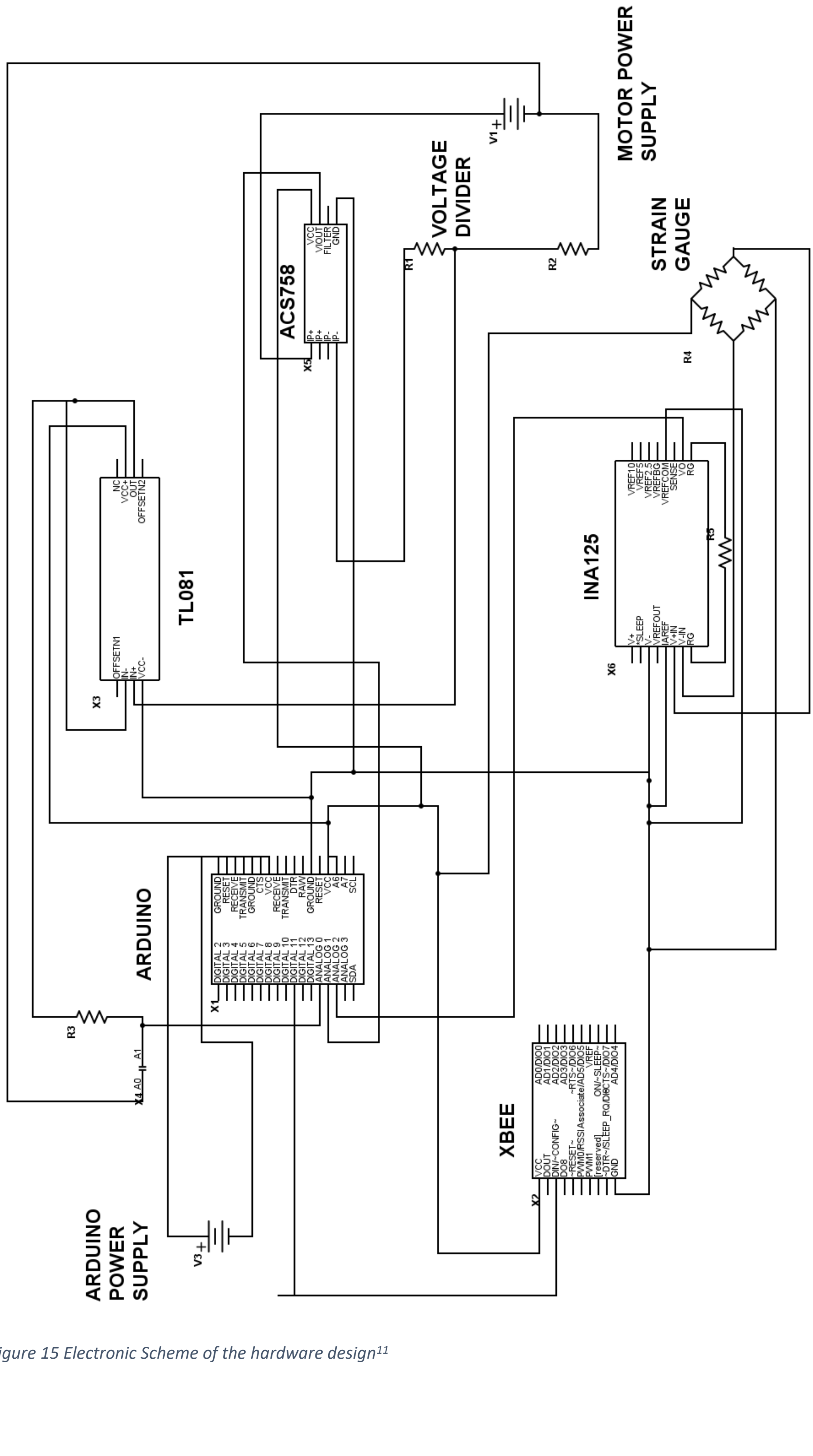


Figure 15 Electronic Scheme of the hardware design[11]

Software design

In order to obtain the output that the contractor needed an evaluation of how the software should work and which software should be used was needed.

The first layer of the software for this project was the Arduino's algorithm. The simplest yet essential of the project. This algorithm would allow the data from the ADC ports of the board to be send by Serial communication to the XBee Transmitter.

The flowchart of the algorithm would be like this:

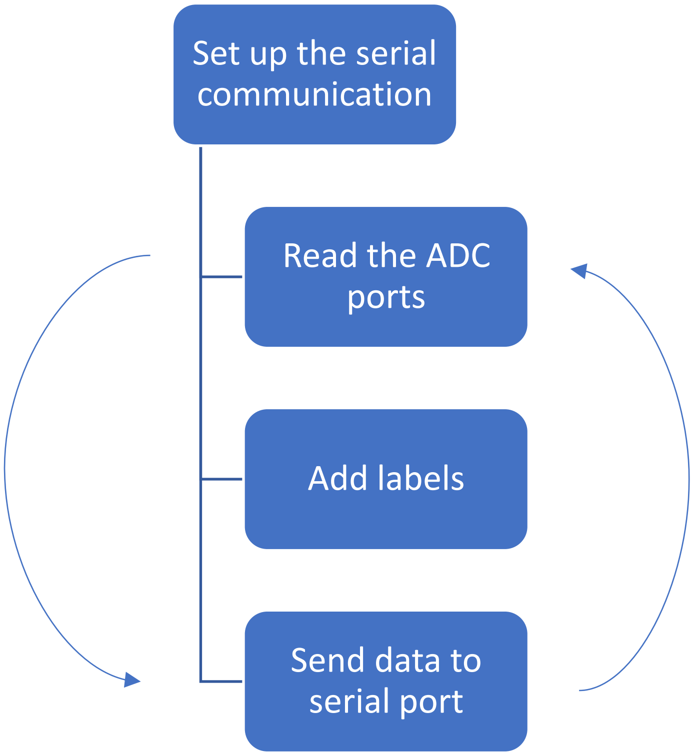


Figure 16 Algorithm of the Arduino Code

The XBee transmitter would send all the data through its protocol to the XBee Receiver, which simultaneously would pass the data received into the USB port in which the user has connected it.

To use the data received on the PC, we needed a programming language that would not intensify the workload, hence easy to learn and code. After the previous research, Python turned out to be the more adequate option due to its open libraries and the amount of information that was free on the internet.

Once the language was chosen, developing the algorithm for the program was necessary.

We needed a program to read from the Serial port in which the XBee explorer was connected and then process this data, since it is read in its digital value directly from the Arduino's ADC, to display it onto a Graphic User Interface for its diagnostic on real time. In addition, the capacity to write that data onto a csv file for further storage and precise analyzation was implemented.

The flowchart of the looped algorithm would be like this:

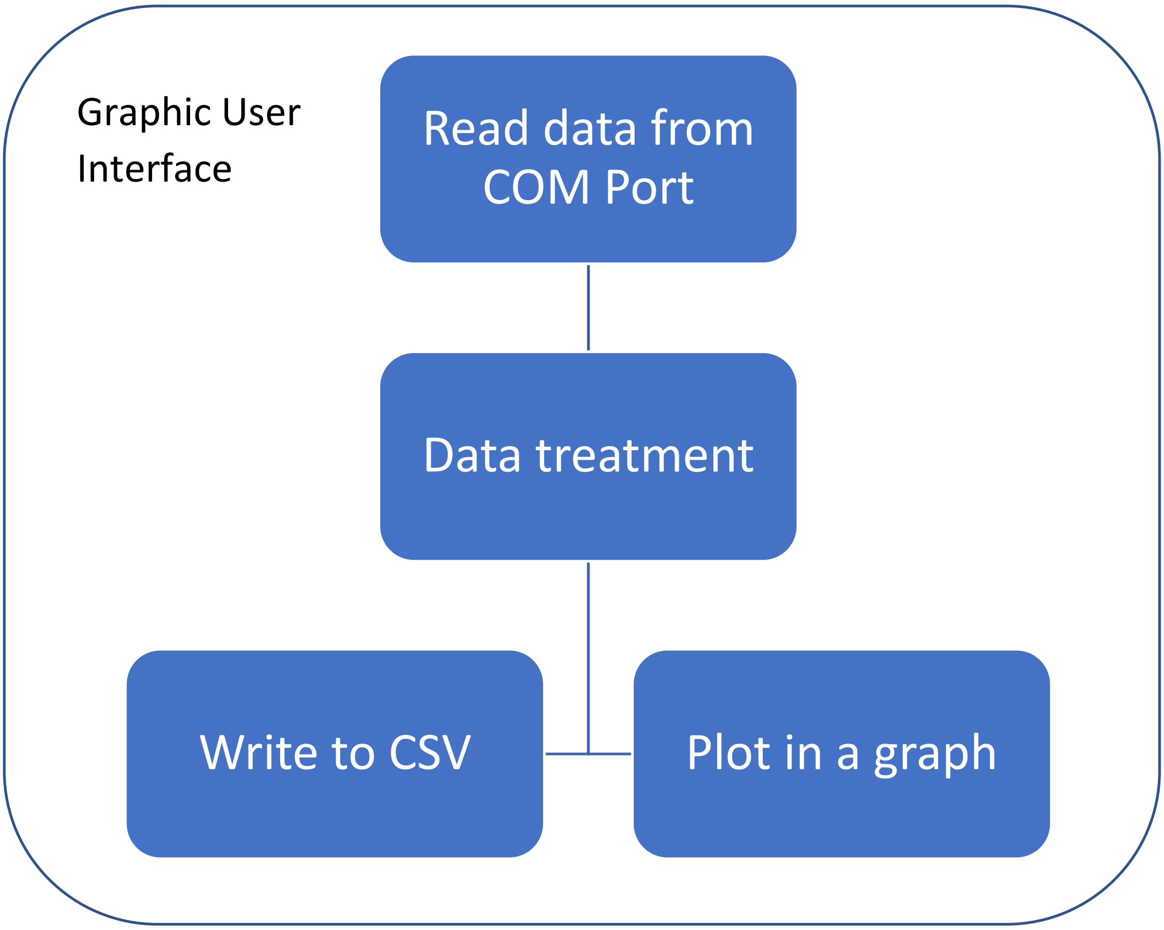


Figure 17 Algorithm for the Python Code

Due to the workaround on the voltmeter, another small algorithm had to be added on the design, now the user would have to enter through the GUI the values of the resistors used on the setup, that information would be used to calculate the digital data received from the COM port in to the real analog values.

This algorithm included in the Data Treatment would be the inverse of the Voltage divider equation plus the conversion of the Digital value to the analog value.

$$Voltage_{system} = Voltage_{read} * \frac{EFSR(\mathrm{Span})}{2^8} * \frac{(R_1+R_2)}{R_2}$$

Libraries Used

The libraries, which the algorithm where designed around are listed and explained exactly like their own documentation.

Matplotlib[12] is a Python 2D plotting library, which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is a thin object-oriented layer on top of Tcl/Tk.

Tkinter is not the only GuiProgramming toolkit for Python. It is however, the most commonly used one.

PIL[13], acronym for **Python Imaging Library** adds image-processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image-processing tool.

Yagmail is a GMAIL/SMTP client that aims to make it as simple as possible to send emails.

Serial This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “serial” automatically selects the appropriate backend.

Time This module provides various time-related functions.

GUI design

The main purpose of a graphic user interface is to display to the user the relevant information and to make it easier for that user to interact with the program or information.

In the scope of this project, a GUI was necessary to represent the data from the sensors in a computer and let the user chose between some action to do regarding supervision, control and data acquisition.

Also being this a custom interface for VUT, it was designed with the logo of the company.

Regarding structure, the main focus should be the plot where the data would be displayed, and the other options being put aside, in a less important place of the frame.

Nevertheless, keeping with the tone of the project, it was never intended to create a fancy or nice looking GUI. The main goal was to be functional and easy to use.

With all those parameters being set the initial draft of the GUI would look somewhat like the fig. 18

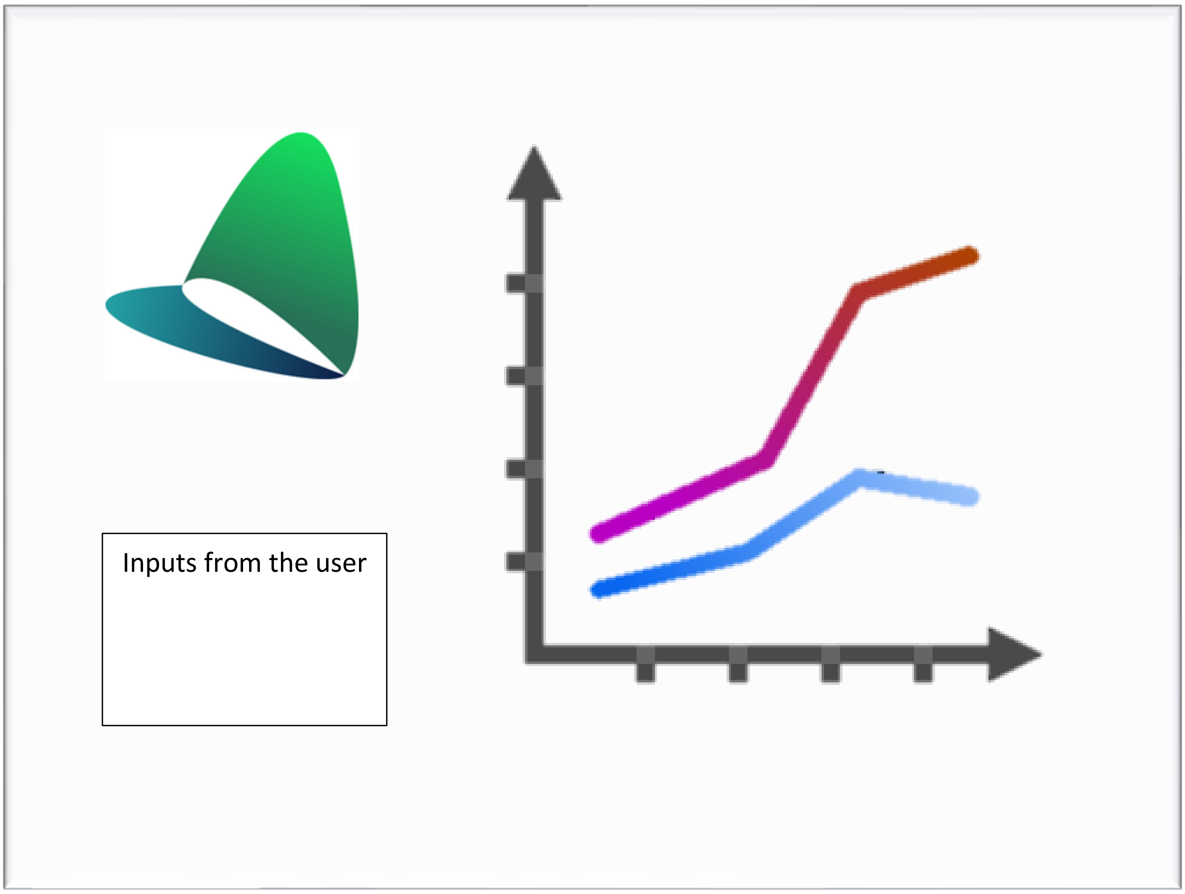


Figure 18 Draft of the design for the GUI

Implementation

In this section it is explained exactly how to reproduce with all the hardware components the prototype, and it is explained the software works for further debug sessions if needed.

For datasheets and the complete source code, see Annex C.

Model implementation

Starting from the first drawings to the more accurate Solidworks designs, the final product was the results from the lessons learned from the mistakes of past structures and was entirely developed with wood, to be lighter and easily repairable.

As shown in the images the testbench was wider and larger than expected due to modifications on the target vehicle that would be mounted on.

Nevertheless, these changes during the development did not affect any of the hardware designs, since they were planned to be upgradable and modular.

The main aspect of the hardware implementation in the model was to keep all electronics inside the main structure for better protection and simpler cable management, instead of the sides of the structure as the design showed.

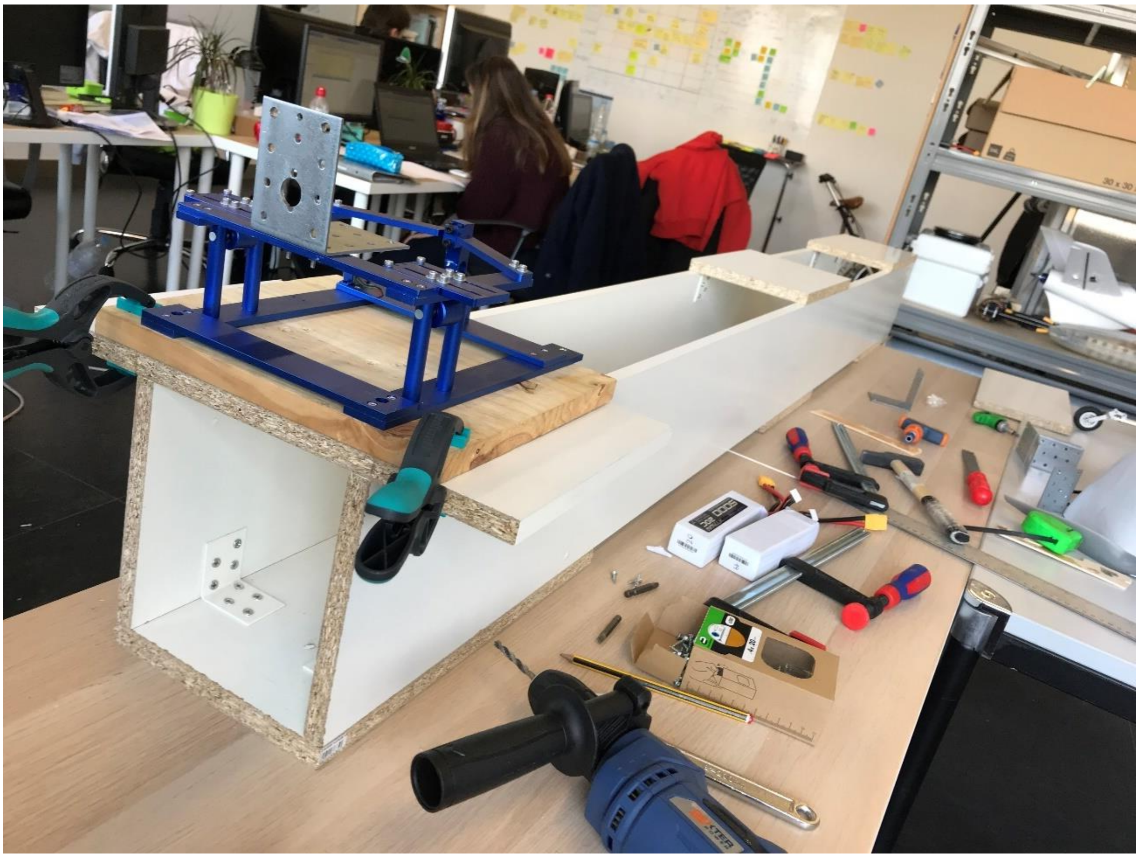


Figure 19 Construction of the structure with wood

In the design phase, a housing for the Arduino and the electronics was planned, but as the voltage divider was used as a workaround, the fact that they should be able to easily change resistors for each unique test concluded that implementing the housing would affect on the user experience, hence it was dismissed.

Now this structure allowed the user to access into the main compartment and modify hardware or just easily remove it and modify outside of the structure.

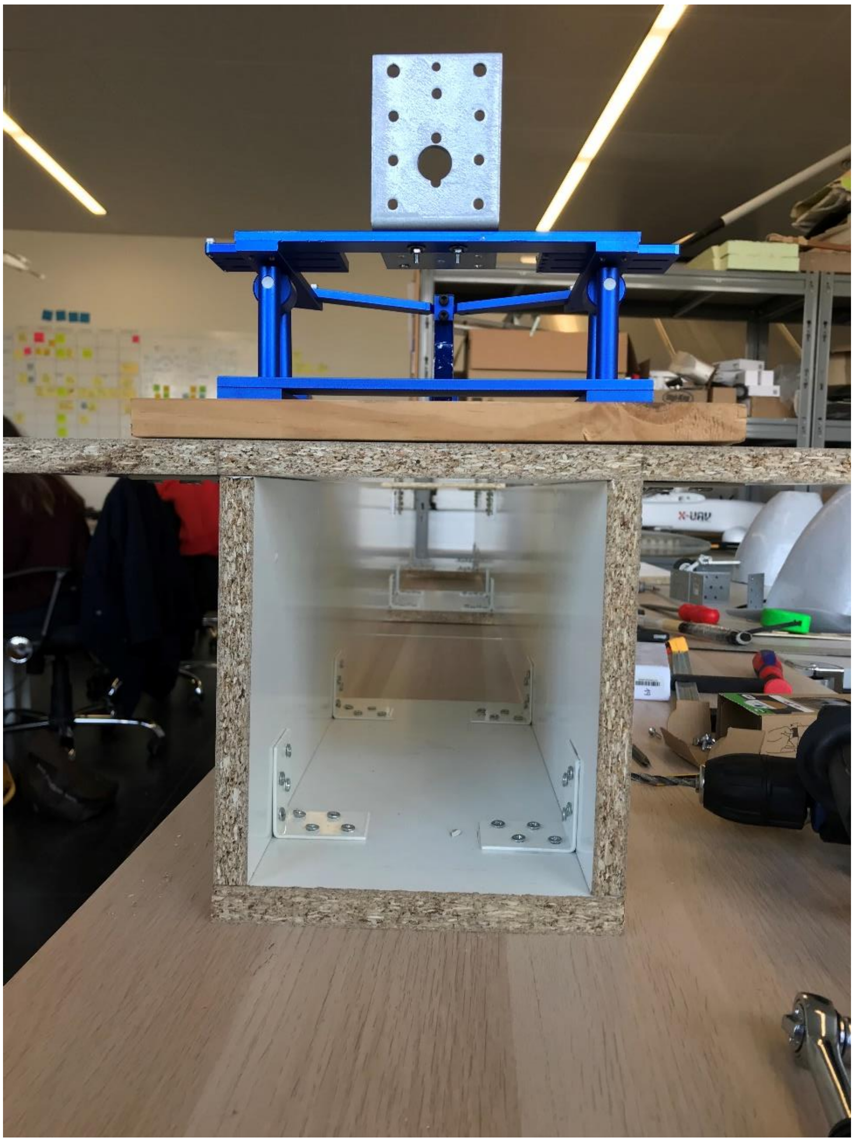


Figure 20 Front view of the structure

Figure 21 Implementation on top of the car

Figure 22 Implementation on top of the car

Hardware implementation

The first task on the implementation of the hardware was the configuration of the 2 XBee modules. In order to do that, the official software of DIGI, the manufacturer, was used, the X-CTU[14].

For two modules of XBee to communicate they have to be in the same network, one has to be a coordinator and the other one an end device, they have to share the same channel and PAN ID.

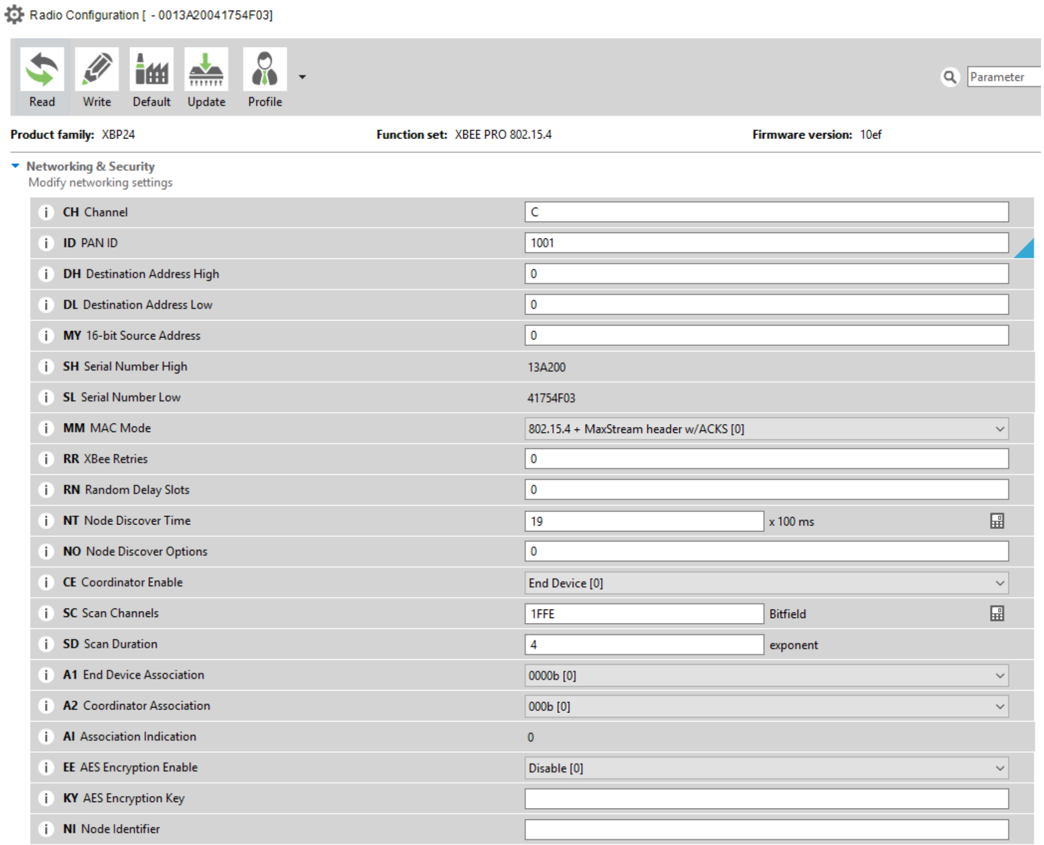


Figure 23 Configuration of the XBee End Device

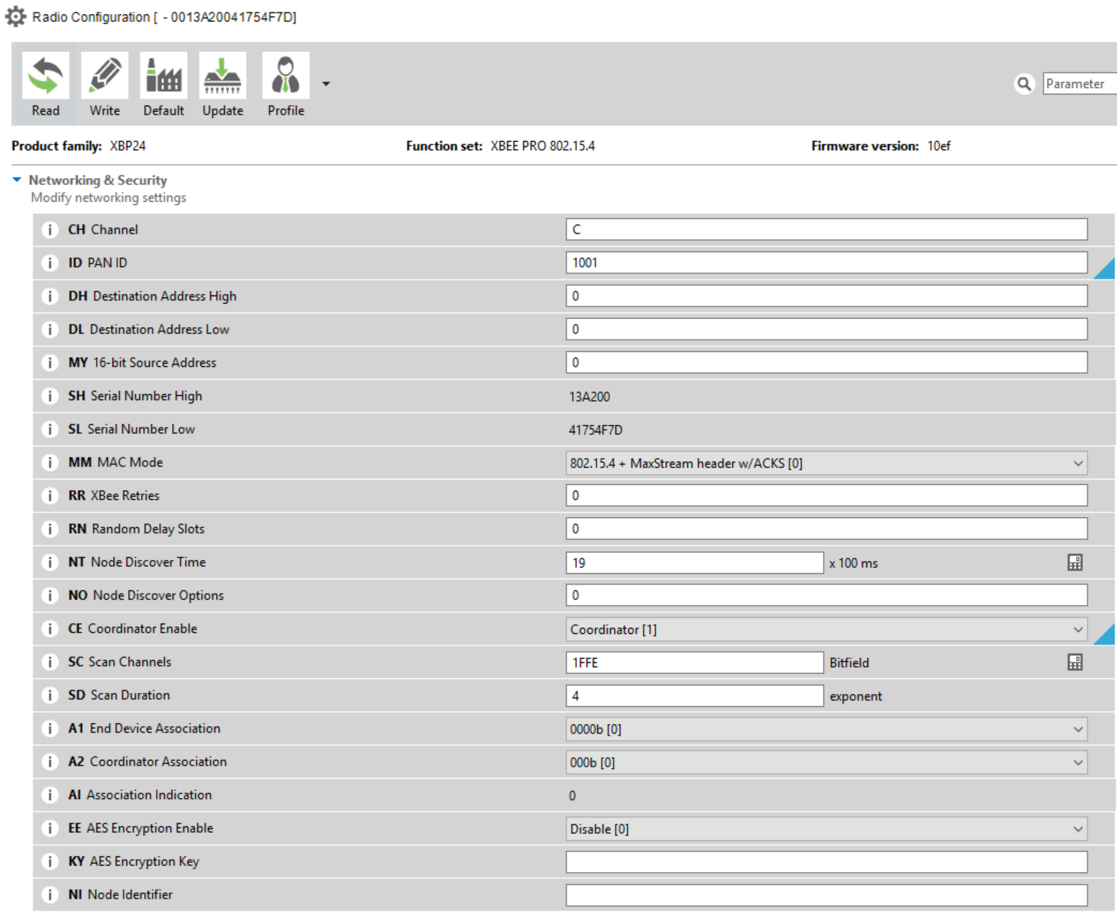


Figure 24 Configuration of the XBee Coordinator

If done correctly, power both of them up, they can see it each other on the XCTU.

With more than one Explorer USB, some debugging and test can be made. For this project, only one Explorer USB fit in the budget, so checking just that both of them couldn’t detect the other one was enough.

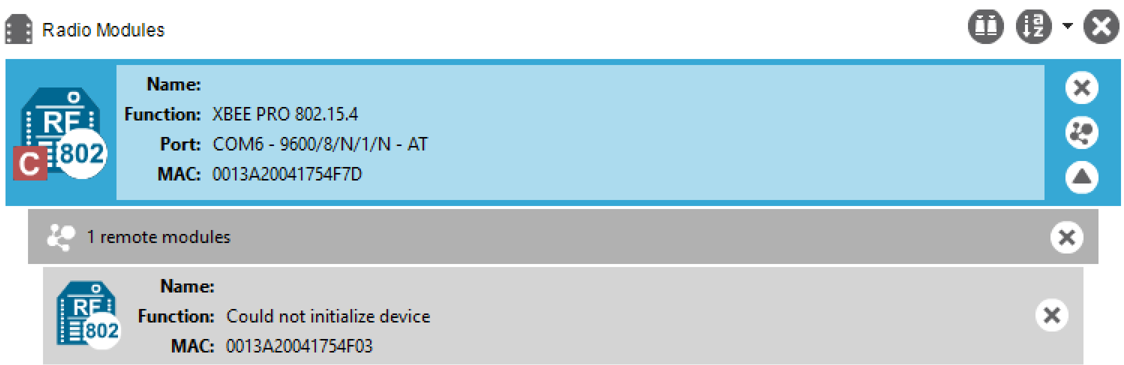


Figure 25 XCTU screenshot of the two modules

With the XBee modules set up, the wiring with the Arduino board could begin.

For the communication between the Arduino and the Coordinator XBee, only 3 cables were needed, the 5 V and ground outputs of the board, and the digital pin 11, which was the output for the data read in the sensors.

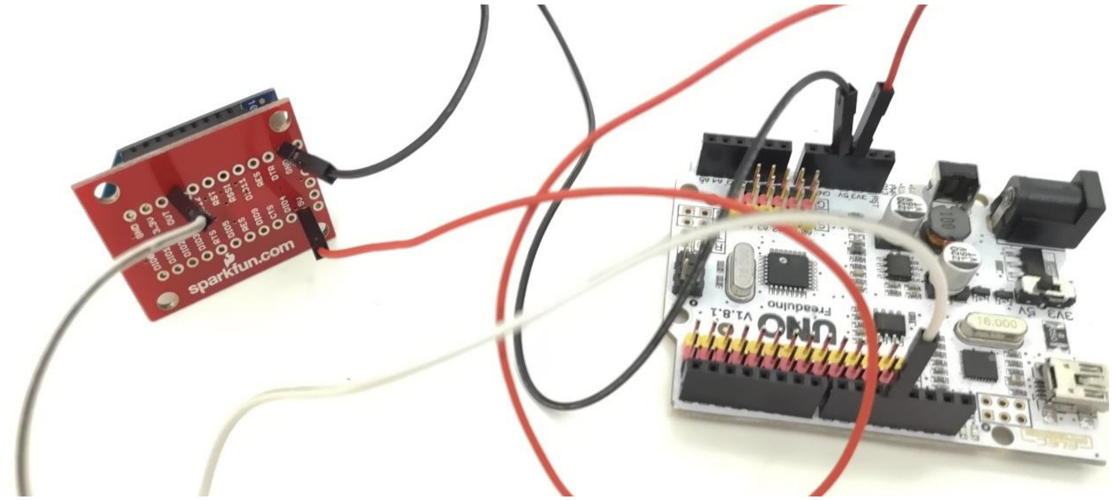


Figure 26 Connection of the XBee Coordinator with the Arduino board

Once having tested the communication sending dummies with the configuration on the Fig.26.

Implementing the rest on the hardware was the next step. Following the scheme on designed on the first phase of the project, the implementation of the final hardware was messy due to the use of breadboards. Nevertheless, as said before, having breadboards allows the user to easily change resistors for recalibrating the sensors or detect failures and replace components.

All the list of components used can be found in the budget section.

Nonetheless, one of the sensors used, the strain gauge needed to be calibrated and have its behavior identified, since VUT did not had any reference nor datasheet.

Using the script listed on the Annex, that only wrote values into a CSV, the strain gauge was calibrated and determined.

$$G = 4 + \frac{60k\Omega}{R_G}$$

Setting the gain to 40 using a Rg of 6kΩ we got 0.12V at 0Kg and 2.4 V at 5 Kg and 3.75 V at 8 Kg

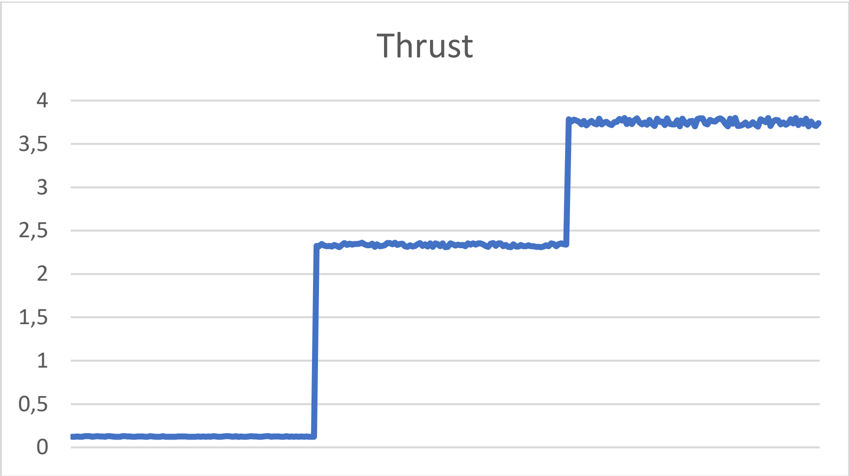


Figure 27 Data gathered on the test for the identification of the strain Gauge

This extrapolated gives us the linearized equation of Y=0.456* X + 0.12.

Figure 28 Linearized equation of the behavior of the strain gauge

From this it can be extracted the zero error and the sensitivity of the gauge.

0.12 V is what needs to be extracted in order to have 0 volts on 0Kg and 0.456 is the linearized sensitivity of the sensor.

For the current sensor some soldering was mandatory, otherwise some noise would be added to the reading due to movement and friction.

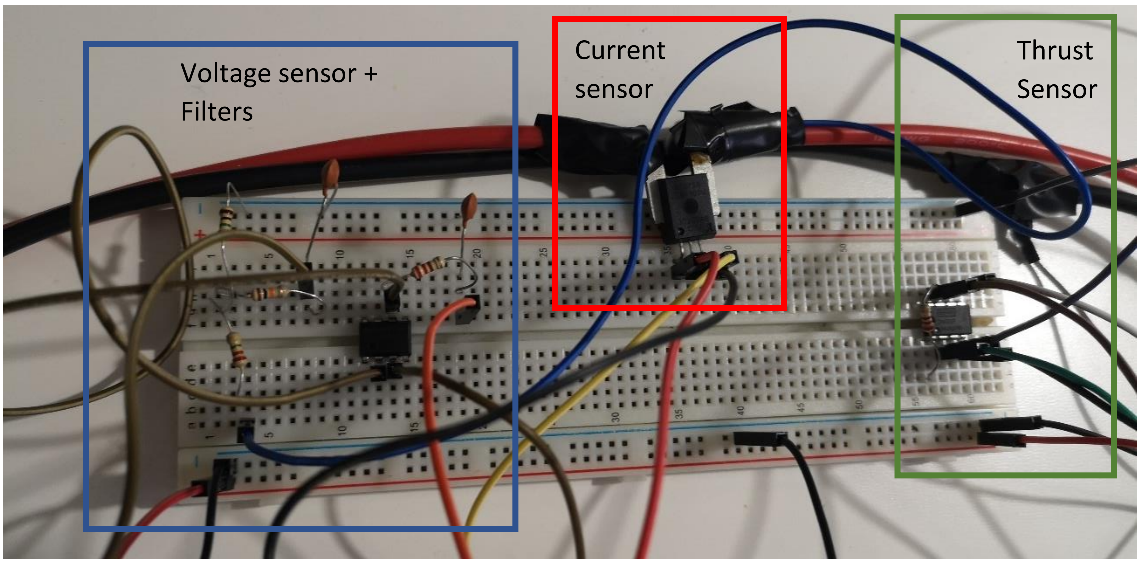


Figure 29 Implementation of the sensors on the breadboard

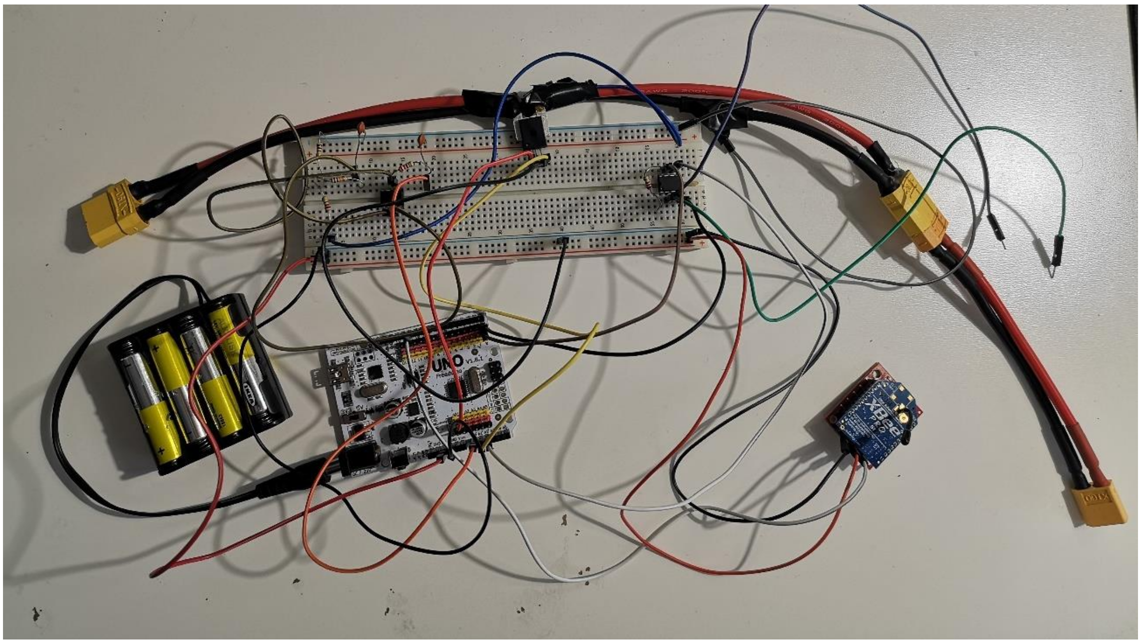


Figure 30 Full set up of the hardware implmentation

In the fig. 30, the high voltage circuit and low voltage circuit are differentiated by the type of cables used, the wider one is the main cable from the power source to the motor.

Software development

The software of the project as any other program, had a frontend, the graphic user interface, and a backend, the data-gathering algorithm.

In order to debug the backend algorithm, a script to simulate the behavior of the hardware was needed to be developed. The script using Com0com software to create virtual serial COM ports, opens a designated COM port, and on a limitless loop, sends random data through it with the same labels as the Arduino does.

Implementing said script in the debugging sessions, was a great advantage, since it allowed hardware free test.

For the backend algorithm, it was considered to use several threads simultaneously. One constantly reading from the COM port, a second one to write on the CSV, every time data was read, and finally the main loop, plotting all the information onto the GUI.

This algorithm was dismissed, when the frontend was implemented because the library used (matplotlib) triggered too many bugs when implemented on a thread. As the schedule was tight a workaround was needed in order to avoid wasting many hours debugging that method.

Instead, the matplotlib animation function was used. This function creates the graph, reads from the COM port, sorts the data by the imposed labels and writes onto the CSV file.

The code implemented followed the structure below:

```
if __name__ == '__main__':

    anim=anim.FuncAnimation(fig, update, frames=60,
interval=350)

    top.mainloop()
```

This is the main of the program, it includes the main loop of the GUI and the call of the function **update**, each 350ms stating the sample time.

```python
def update(i):

    data_raw=[0]
    global counter
    global data
    global data2
    global VR2
    global VR1
    VR1=0
    VR2=0
    alarms_activated = False

    if CheckVar2.get()==1 :
        setup= True
        VR2=int(R2.get())
        VR1=int(R1.get())

    elif CheckVar2.get() == 0:
        setup= False
        first.lift()
    incomingByte = ser.read().decode('utf-8')

    if CheckVar1.get()==1:
        alarms_activated = True

    elif CheckVar1.get() == 0:
        alarms_activated = False

    if incomingByte =='<':
        func = ser.read().decode('utf-8')
        if func== 'v':
            v(incomingByte,alarms_activated,setup,VR1,VR2)
        elif func=='t':
            t(incomingByte,alarms_activated,setup)
        elif func=='c' :
            c(incomingByte,alarms_activated,setup)
```

This is the function **update**, it handles the change of the graph every time it is called.

Initializes the variables that will be passed as arguments onto the three different subfunctions. Checks whether the user has activated the alarms for the thresholds or not and gets the values for the voltage divider.

Finally reads the first byte of the buffer of the COM port and depending on the character read calls the function that correspond on each magnitude.

The subfunction in charge of the voltage follows this structure.

```
def v(incomingByte,alarms_activated,setup,VR1,VR2 ):
```

It receives as arguments the variables to set up the alarm and the parameters of the Voltage divider from the function **update**.

The most important argument is the **incomingByte**, which is the cue to start reading the values.

```
        data_raw_v=[0]
        global counter_v
        global data_v
        global data2_v
        while incomingByte != '>':
```

While the character read and stored in the variable **incomingByte** is not >, it means we still reading valuable data, hence the loop.

```
            incomingByte = ser.read().decode('utf-8')
```

For each iteration of this loop we store in **incomingByte** another character and treat it depending on what it is.

```
            if incomingByte == '>':
```

If it is “>”, it means the transmission of the particular value we were reading is over and we shall continue reading others, hence the instruction **pass**.

```
                Pass
```

Otherwise we add the value read to a list with the other characters.

```
            else:
                data_raw_v.append(incomingByte)

        num = int(''.join(map(str,data_raw_v)))
```

When the while is over and we have iterated from “<” to “>”, the transmission is completed, meaning all the data is now stored in the variable **num.**

Now that we have all the numeric data in a variable we can start using it to analyze and plot.

```
        if alarms_activated == True:
            if  num <= int(threshold_voltage.get()):
```

If the value is higher than the threshold introduced from the GUI and the alarms are activated, an email is send to the desired email address to notify the admin that the threshold has been surpassed.

```
                print("Threshold for Voltage surpassed!")
                yag = yagmail.SMTP('venturiutech@gmail.com',
'Password')
                yag.send('alertmail@gmail.com', "Alert",
"Threshold for Voltage surpassed!"+'\n'+"Time:
"+time.asctime(time.localtime())+'\n'+ "Value Reached:
"+threshold_voltage.get())
```

If the user has not introduced the values from the voltage divider, as the data would make no sense, since it would be only in the range from 0-5v, it won't be displayed
If the user has introduced the data, then as the other functions, it plots the data into the canvas and represents it on the GUI.

```
        if setup==True:
```

Conversion from digital to analog again to read the actual value of the magnitude.

```
            num= num*5/1024
```

Conversion from the output of the voltage divider to the input.

```
            num= num*(VR1+VR2)/VR2
```

Simultaneously the value is written onto the CSV file for further analysis and storage.

```
            f.write('Voltage:')
            f.write(str(num)+'\n')
```

Finally, we append the value read (num) and the time into separate list in order to plot them as X and Y axis on a graph.

```
            data_v.append(num)

            data2_v.append(time.asctime(time.localtime()))

            line, = ax1.plot( data2_v,data_v, color='blue',
lw=2)
            plt.draw()
            counter_v+=1

            data_raw_v=[]
```

When the counter has reached the value of 25, which is arbitrary, just not to oversaturate the graph, it clears the axis and the counter to start the sequence of plotting again.
Note that all the actual data will be stored on the CSV, so no data would be lost.

```
if counter_v > 25 :

   counter_v=0
   ax1.clear()
   ax1.set_ylabel('Voltage')
   plt.xticks(rotation=50)
   data_v = []
   data2_v = []
```

The only changes in the structure for the other two functions are the treatment of the data read.

For the subfunction in charge of current:

Conversion from digital to analog again to read the actual value of the magnitude.

```
num = num * 5/1024
```

Conversion to the actual value of current taking into account the offset of the sensor which is 0,12 * VCC, and the sensitivity which is 40 mV.

```
num = (num - 0.656)/0.04
```

For the subfunction in charge of thrust:

Conversion from digital to analog again to read the actual value of the magnitude.

```
num = num * 5/1024
```

Conversion to the actual value taking into account the zero error and the sensitivity of the sensor.

```
num = (num - 0.12) * 3.2
```

Overall, our backend and frontend compatibility forced the system to a constant sample time of 330 ms for each magnitude to be read.

In terms of the GUI, several layouts were discussed with VUT, since a user-friendly program was the main goal of this end of the software development.

We decided to have 3 plots, sharing the X axis, and separate Y axis hence each magnitude had its own thresholds and range of values. Sharing also the Y axis, which may increase the aesthetics of the interface lowered the resolution of the graph. Utility over aesthetic was chosen.

The GUI also incorporated an image from the company's logo to add a more personal touch, as well as the buttons to stop the program, open the CSV, and the tool to calculate a new set of voltage divider for the voltmeter.

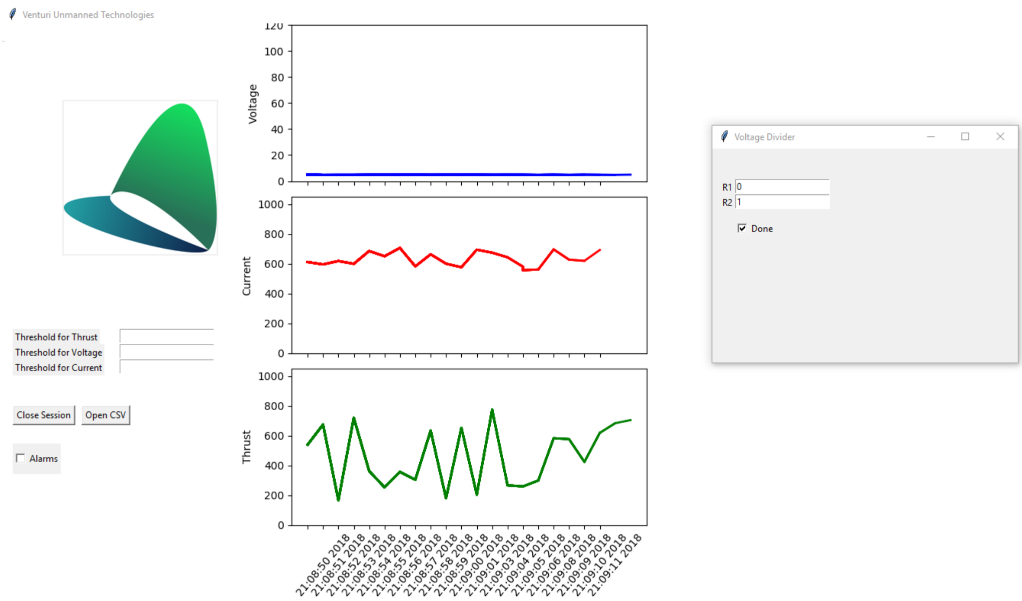


Figure 31 GUI of the program

Results and Evaluation

Budget

In this section it is listed the direct cost of the components used in the prototype as well as the material and the indirect cost of the time of the junior engineer in charge of the project and the senior engineer in charge of supervising it.

Component	Unit price (€)	Quantity	Total cost
XBee S1 Pro	46,6	2	**93,2**
XBee Explorer USB	25,4	1	**25,4**
XBee Breadboard adatpter	15,4	1	**15,4**
Arduino UNO	32,67	1	**32,67**
BreadBoard	8,6	1	**8,6**
INA125	5,49	1	**5,49**
TL081	0,47	1	**0,47**
ACS758	7,15	1	**7,15**
Strain Gauge	7,2	1	**7,2**
Capacitors 100µF	0,3	2	**0,6**
Structure of wood		-	**40**
Arduino's Power Supply	5,6	1	**5,6**
Set of Resistors	6,1	1	**6,1**
Total cost			**247,88**

Table 2 Direct costs of the components

A total cost of 240,68 Euros for the components needed for the construction of the prototype.

Task	Hourly rate(€)	Quantity	Total cost(€)
State of the art research	12	10	**120**
Kick Off meeting with VUT	12	4	**48**
Design			
Model Design	15	15	**225**
Software Design	15	20	**300**
Hardware Design	15	20	**300**
Implementation (Includes laboratory hours)			
Model Implementation	15	10	**150**
Software Implementation	15	30	**450**
Hardware Implementation	15	20	**300**
Test and Evaluation			
Prototype Test #1	15	6	**90**
Prototype Test #2	15	6	**90**
Evaluation	12	4	**40**
Documentation	12	25	**300**
Senior Engineer Supervising	25	6	**2353**
Total cost			**4646**

Table 3 Indirect costs of the worktime of the enginners

The in direct cost of the project adds up to 4646 Euros.

Direct Cost	**247,88 €**
Indirect Cost	**4646 €**
Total	**4893,88 €**

Table 4 Total cost of the project

The total cost of the project is 4893,88 Euros.

Prototyping

Indoor tests

A test with the first working hardware was done in order to detect bugs or malfunction in an earlier state of implementation.

For that test it was only measured current and voltage and the only funtionality that was implemented was writting to a CSV. Find the code used attached in the Annex.

The hardware setup consisted on the already working voltage divider as a voltmeter and the current sensor. The signals from those two sensors directly into the ADC's of the Arduino and send to the XBee network.

The results on this test revealed the requirement of a filter as shown in the Fig 32, a lot of noise was broght in the data read.

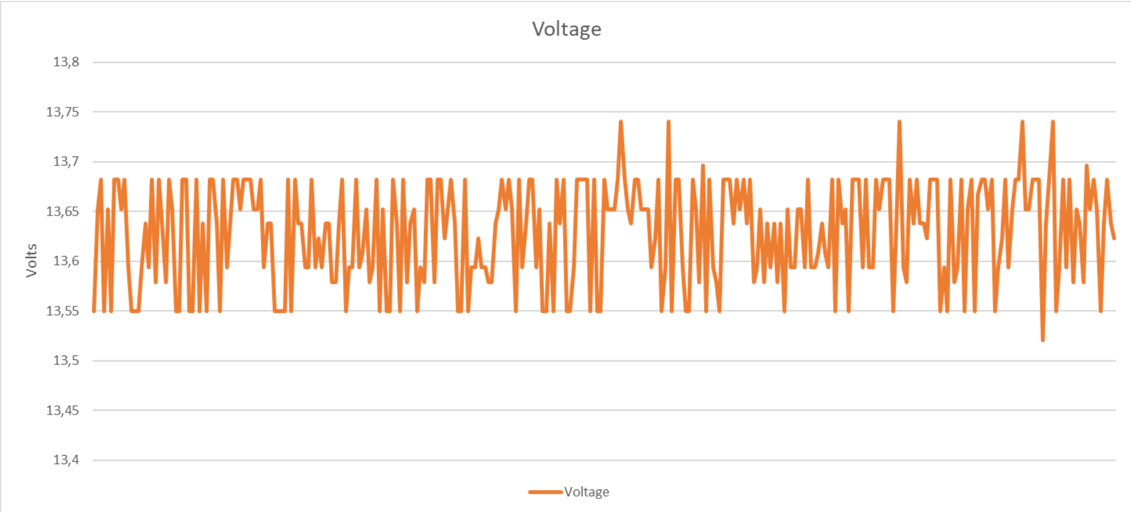


Figure 32 Data gathered on the first test, Voltage

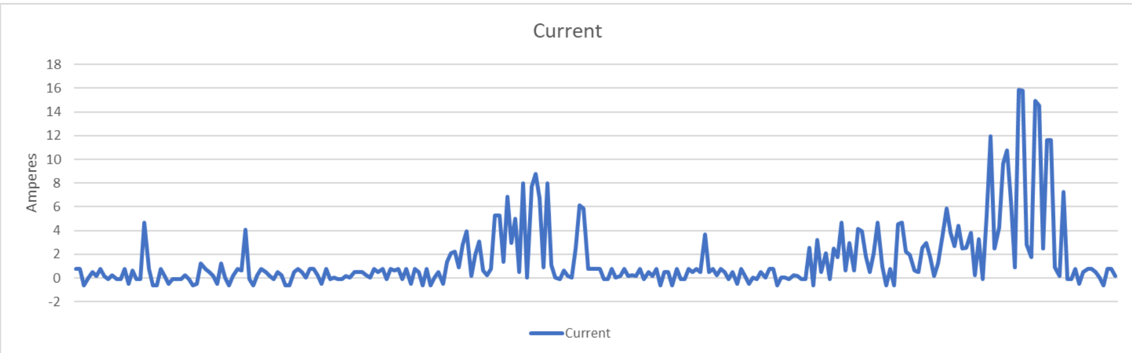


Figure 33 Data gathered on the first test, Current

As the results for this test followed the expected, the implementation moved on a better software, graphical visualization and implementing the strain gauge on the hardware.

Due to conflict of schedules from both parties, there was only another test, which would be the last before the final release of the prototype.

On that test the final version of the software was used, as well as all the hardware implemented on the bench.

The main outputs of the test were that the new GUI and the algorithm to make it work, added some delay on the data. Nevertheless, using the filters, the noise from the thrust and voltage were attenuated but the data of the current continued to show the particular signal wave form expected from a current on a PWM system.

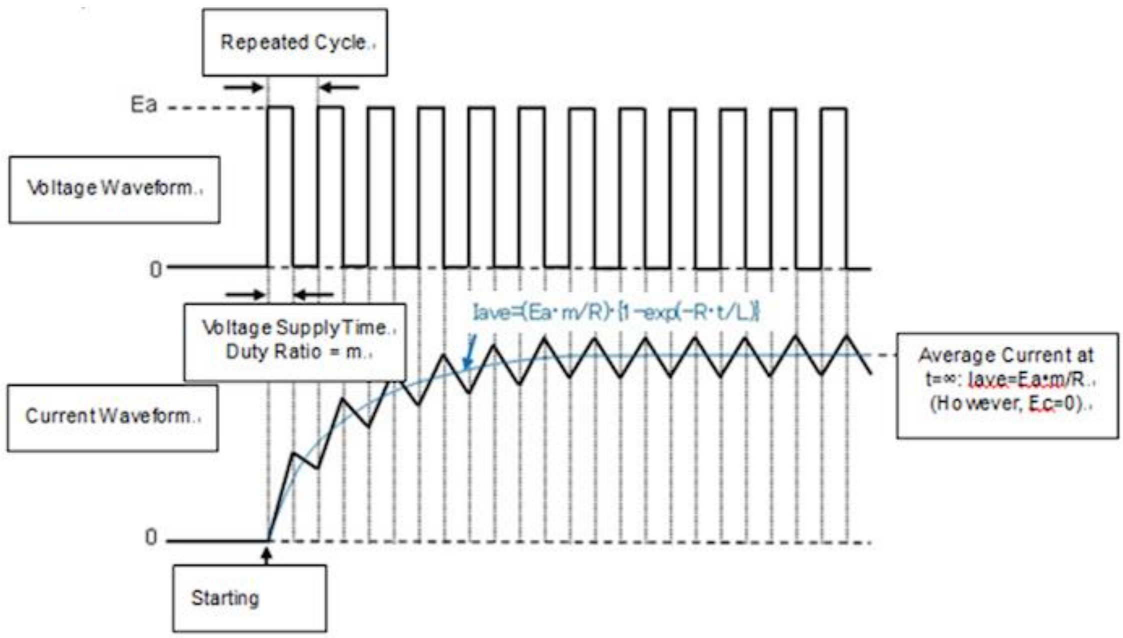


Figure 34 Representation of the voltage and current waveforms on a PWM

Outdoor test

The outdoor test was scheduled after the due date for this dissertation. Nonetheless, the expected output from an outdoor test in the current state of the prototype is a substantial different data, since the power source used on a real test is higher than the indoor test, but no other effects should appear, since the integration of the hardware inside the structured is designed to guarantee its functionality.

Conclusions

To conclude the project after the evaluation of the system, the final verdict is that the systems works as expected. It is not the best solution, due to low budget, nonetheless, it lets the system open to upgrades and further improvements.

Looking back to the solution VUT had before this project began, a GoPro attached to a structure to record the values displayed in the LCD of the thrust stand they had at the moment, this solution solves many of the problems.

In addition, automatically stores all the data to a CSV, there is no need for a person to check all the videos the GoPro recorded anymore to see if any threshold was surpassed and write it down.

Even though the system adds some delay on the data read, it is still a working prototype that solves the main problem and offers an initial solution.

In order to improve this project and have in the future a final product that could be used as an asset for the company, in this dissertation all the main points of the design and implementation have been tediously explained, so it can be reproduced and understood by any other engineer.

By all means, this project not only fulfilled the requirements of VUT, but also created a bond of mutual learning from both sides, the aerospace engineers from VUT learned about the electronics of the system being delivered to them and they shared the experience of being part of the development of their start-up.

It is a valuable output to learn skills beside technical, like design and time management from entrepreneurs, from a Bachelor's Final Degree Project.

Future Work

In this section, all the improvements that could be applied to the project to improve the prototype will be explained. All these possible upgrades were not implemented due to time, budget or simply because they were discovered during the prototyping phase.

A soft improvement would be making the software more solid and faster, creating a personal python library special for the solution in particular, the software developed in this project was entirely made by one engineer, that just started working on Python. Further study of the language and the libraries used could improve the overall performance of the code and add up to a better sample time and a more user-friendly GUI.

Add more sensors or improve the ones being used with higher quality components.

Embed the system for a more aesthetic and pleasing effect on the implementation into the structure. Create a custom case that still allows to easily access the hardware but keeping in more organized.

Increase the complexity of the XBee system, by adding XBee with 3G capabilities and storage the data into an online database directly, using a XAMP platform.

Webography

1. https://www.digi.com/xbee
2. https://www.rcbenchmark.com/dynamometer-series-1580/
3. http://www.turnigy.com/
4. https://www.arduino.cc/
5. https://www.python.org/
6. https://wiki.python.org/moin/TkInter
7. http://com0com.sourceforge.net/
8. http://www.solidworks.es/sw/3d-cad-design-software.htm
9. https://www.allegromicro.com/~/media/Files/Datasheets/ACS758-Datasheet.ashx
10. http://www.ti.com/lit/ds/symlink/ina125.pdf
11. https://www.schematics.com/editor/
12. https://matplotlib.org/
13. https://pillow.readthedocs.io/en/5.1.x/
14. https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

Annex A: Code

Codigo Arduino

```
#include <SoftwareSerial.h>

// software serial #1: RX = digital pin 10, TX = digital pin 11
SoftwareSerial portOne(10, 11);
int voltage_port = A0;
int current_port = A1;
int thrust_port = A2;

int value1= 0;
int value2=0;
int value3=0;
String Label1= "v";
String Label2= "c";
String Label3= "t";

void setup() {



  // Start each software serial port
  portOne.begin(9600);
 Serial.begin(19200);
}

void loop() {
int value1= analogRead(voltage_port);
int value2=analogRead(current_port);
int value3=analogRead(thrust_port);

Serial.print("<"+Label1+value1+">");
portOne.print("<"+Label1+value1+">");
delay(100);

Serial.print("<"+Label2+value2+">");
portOne.print("<"+Label2+value2+">");
delay(100);
Serial.print("<"+Label3+value3+">");
portOne.print("<"+Label3+value3+">");
delay(100);

}
```

Script of the first test

```
import serial
from time import sleep
import threading


import csv
msg=[]

class myThread (threading.Thread):
   def __init__(self):
      threading.Thread.__init__(self)

   def run(self):

        read_comport()

      # Get lock to synchronize threads
      # threadLock.acquire()

      # Free lock to release next thread
      # threadLock.release()

def read_comport ():

    ser = serial.Serial(
        port='COM3' ,\
        baudrate=9600,\
        parity=serial.PARITY_NONE,\
        stopbits=serial.STOPBITS_ONE,\
        bytesize=serial.EIGHTBITS,\
            timeout=0)

    print("connected to: " + ser.portstr)
    index=0
    msg=[]
    #this will store the line

    while ser.read()!= '\0':
        incomingByte = ser.read().decode('utf-8')

        if incomingByte == '<':
            while incomingByte != '>':

                incomingByte = ser.read().decode('utf-8')
                if incomingByte == '>':
                    pass
                else:
                    msg.append(incomingByte)

            print('Com4:'+''.join(msg))
            # toWrite=''.join(msg)

            f.write(''.join(msg)+'\n')
            msg=[]
            sleep(0.1)
        # elif index < 10 :
```

```
            # msg[index] = incomingByte
            # index += 1

        # using ser.readline() assumes each line contains a single reading
        # sent using Serial.println() on the Arduino
        # reading = ser.readline().decode('utf-8')
        # reading is a string...do whatever you want from here

    ser.close()


if __name__ == '__main__':
    f = open('csvfile.csv','w')

    read = myThread()
    read.start()
    read.join()
```

Write Random.PY

```
import serial
import random
from time import sleep

start="<"
end=">"
voltage="v"
current="c"
thrust="t"

ser = serial.Serial(
    port='COM7',\
    baudrate=9600,\
    parity=serial.PARITY_NONE,\
    stopbits=serial.STOPBITS_ONE,\
    bytesize=serial.EIGHTBITS,\
        timeout=0)

print("connected to: " + ser.portstr)

while True:
    ser.write(start.encode('utf-8'))
    ser.write(voltage.encode('utf-8'))
    ser.write(str(int(random.uniform(980, 1024))).encode('utf-
8'))
    ser.write(end.encode('utf-8'))
    ser.write(start.encode('utf-8'))
    ser.write(current.encode('utf-8'))
    ser.write(str(int(random.uniform(550, 724))).encode('utf-
8'))
    ser.write(end.encode('utf-8'))
    ser.write(start.encode('utf-8'))
    ser.write(thrust.encode('utf-8'))
    ser.write(str(int(random.uniform(150, 778))).encode('utf-
8'))
    ser.write(end.encode('utf-8'))


ser.close(
```

Source code with comments

```
#!/usr/bin/python3
from tkinter import *
import tkinter
import yagmail
import tkinter.messagebox
from time import sleep
import os
from PIL import ImageTk, Image
import serial
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt
import matplotlib.animation as anim
import time


top = tkinter.Tk()
top.title("Venturi Unmanned Technologies")
top.geometry("1920x1080")
top.configure(background='white')
f = open('csvfile.csv','w')

first=tkinter.Toplevel(top)
first.title("Voltage Divider")
first.geometry("400x280")
L1 = Label(top,text="Threshold for Thrust")
L1.place(y=400, x=10)

L2 = Label(top,text="Threshold for Voltage")
L2.place(y=420, x=10)

L3 = Label(top,text="Threshold for Current")
L3.place(y=440, x=10)

L4 = Label(first,text="R1")
L4.place(y=40, x=10)

L5 = Label(first,text="R2")
L5.place(y=60, x=10)
global R1
global R2
R1=Entry(first)
R1.place(y=40, x=30)

R2=Entry(first)
R2.place(y=60, x=30)

threshold_thrust=Entry(top)
```

```
threshold_thrust.place(y=400, x=150)

threshold_voltage=Entry(top)
threshold_voltage.place(y=420, x=150)

threshold_current=Entry(top)
threshold_current.place(y=440, x=150)

global alarms_activated
alarms_activated = True
global setup
setup=False
global incomingByte
def v(incomingByte,alarms_activated,setup,VR1,VR2 ):
    # We receive as arguments the variables to set up the alarm
and the parameters of the Voltagdivider
    # The most important argument is the incomingByte, which is
the cue to start reading the values.
        data_raw_v=[0]
        global counter_v
        global data_v
        global data2_v
        while incomingByte != '>':
            # Meanwhile the carachter read and stored in the
variable Incomingbyte is not >, it means we still reading
valuable data, hence the loop.
            incomingByte = ser.read().decode('utf-8')
            # For each iteration of this loop we store in
incomingByte another caracther and treat it depending on what it
is
            if incomingByte == '>':
            # If it is >, it means the transmission of the
particular value we were reading is over and we shall continue
reading others, hence the instruccion pass
                pass
            else:
            # Otherwise we add the value read to a list with the
other carachters
                data_raw_v.append(incomingByte)

        num = int(''.join(map(str,data_raw_v)))
        # When the while is over and we have iterated from < to
>, the transimission is completed, meaning all the data is now
stored in the variable num

        # Now that we have all the numeric data in a variable we
can start using it to analyse and plot.
        if alarms_activated == True:
            if  num <= int(threshold_voltage.get()):
```

```
            # If the value is higher than the threshold
introduced from the GUI and the alarms are activated, a email is
send to the desired email address to notify the admin that the
threshold has been surpassed.
                print("Threshold for Voltage surpassed!")
                yag = yagmail.SMTP('venturiutech@gmail.com',
'EduardTFG')
                yag.send('eduardmgobierno@gmail.com', "Alert",
"Threshold for Voltage surpassed!"+'\n'+"Time:
"+time.asctime(time.localtime())+'\n'+ "Value Reached:
"+threshold_voltage.get())

        if setup==True:
        # If the user has not introduced the values from the
voltage divider, as the data would make no sense, since it would
be only in the range from 0-5v, it won't be displayed
        # If the user has introduced the data, then as the other
functions, it plots the data into the canvas and represents it
on the GUI.
        # Conversion from digital to analog again to read the
actual value of the magnitude
        # Conversion from the output of the voltage divider to
the input
            num= num*5/1024
            num= num*(VR1+VR2)/VR2

         # Simultaneously the value is written onto the CSV file
for further analysis and storage
            f.write('Voltage:')
            f.write(str(num)+'\n')

        # Finally we append the value read (num) and the time
into separate list in order to plot them as X and Y axis on a
graph
            data_v.append(num)

            data2_v.append(time.asctime(time.localtime()))

            line, = ax1.plot( data2_v,data_v, color='blue',
lw=2)
            plt.draw()
            counter_v+=1

            data_raw_v=[]

        # When the counter has reached the value of 25, which is
arbitary, just not to oversaturate the graph, in clears the axis
and the counter to start the sequence of plotting again
```

```
        # Note that all the actual data will be stored on the
CSV, so no data would be lost.
        if counter_v > 25 :

           counter_v=0
           ax1.clear()
           ax1.set_ylabel('Voltage')
           plt.xticks(rotation=50)
           data_v = []
           data2_v = []

def c(incomingByte, alarms_activated,setup):

        data_raw_c=[0]
        global counter_c
        global data_c
        global data2_c

        while incomingByte != '>':

            incomingByte = ser.read().decode('utf-8')
            if incomingByte == '>':
                pass
            else:

                data_raw_c.append(incomingByte)

        num = int(''.join(map(str,data_raw_c)))

        if alarms_activated == True:
            if num >= int(threshold_current.get()) :
                print("Threshold for Current surpassed!")
                yag = yagmail.SMTP('venturiutech@gmail.com',
'EduardTFG')
                yag.send('eduardmgobierno@gmail.com', "Alert",
"Threshold for Current surpassed!"+'\n'+"Time:
"+time.asctime(time.localtime())+'\n'+ "Value Reached:
"+threshold_current.get())

        if setup==True:
            # Conversion from digital to analog again to read
the actual value of the magnitude
            num = num * 5/1024
            # Conversion to the actual value of current taking
into account the offset of the sensor which is 0,12 * VCC, and
the sensitivity which is 40 mV
            num = (num - 0.656)/0.04
            f.write('Current:')
            f.write(str(num)+'\n')
```

```
            data_c.append(num)

            data2_c.append(time.asctime(time.localtime()))

            line, = ax2.plot( data2_c,data_c, color='red', lw=2)
            plt.draw()
            counter_c+=1

            data_raw_c=[]

            if counter_c > 25 :

               counter_c=0
               ax2.clear()
               ax2.set_ylabel('Current')
               plt.xticks(rotation=50)
               data_c = []
               data2_c = []

def t(incomingByte, alarms_activated,setup):

        data_raw_t=[0]
        global counter_t
        global data_t
        global data2_t
        while incomingByte != '>':

            incomingByte = ser.read().decode('utf-8')
            if incomingByte == '>':
                pass
            else:

                data_raw_t.append(incomingByte)

        num = int(''.join(map(str,data_raw_t)))

        if alarms_activated == True:
            if num >= int(threshold_thrust.get()):
                print( alarms_activated)
                print("Threshold for Thrust surpassed!")
                yag = yagmail.SMTP('venturiutech@gmail.com',
'EduardTFG')
                yag.send('eduardmgobierno@gmail.com', "Alert",
"Threshold for Thrust surpassed!"+'\n'+"Time:
"+time.asctime(time.localtime())+'\n'+ "Value Reached:
"+threshold_thrust.get())


        if setup==True:
```

```
            # Conversion from digital to analog again to read
the actual value of the magnitude
            num = num * 5/1024
            # Conversion to the actual value taking into account
the zero error and the sensivity of the sensor
            num = (num - 0.12) * 3.2
            f.write('Thrust:')
            f.write(str(num)+'\n')
            data_t.append(num)

            data2_t.append(time.asctime(time.localtime()))

            line, = ax3.plot( data2_t,data_t, color='green',
lw=2)
            plt.draw()
            counter_t+=1

            data_raw_t=[]

            if counter_t > 25 :

               counter_t=0
               ax3.clear()
               plt.xticks(rotation=50)
               ax3.set_ylabel('Thrust')
               data_t = []
               data2_t = []

def endloop():

    sys.exit()
# This functions allows the program to open the csvfile where
all the data has been stored while the process was running.
# In order to the data, stored on the RAM of the pc, to be
written in the CSV, the file has to be closed and the animation
stopped.
def opencsv():
    anim.event_source.stop()
    f.close()
    os.system('csvfile.csv')


top.protocol('WM_DELETE_WINDOW',endloop)
# Those are the two buttons that allow the user to interact,
either stop the streaming and open the CSV or just finish the
session.
b1 = tkinter.Button(top, text="Close Session",command=endloop)
b1.place(y=500, x=10)
b2 = tkinter.Button(top, text="Open CSV",command=opencsv)
```

```
b2.place(y=500, x=100)



global CheckVar1
CheckVar1 = IntVar()
C1 = Checkbutton(top, text = "Alarms", variable = CheckVar1, \
                 height=2, \
                 width = 5)
C1.place(y=550, x=10)

global CheckVar2
CheckVar2 = IntVar()
C2 = Checkbutton(first, text = "Done", variable = CheckVar2, \
                 height=2, \
                 width = 5)
C2.place(y=85, x=25)

img = ImageTk.PhotoImage(Image.open("0.png"))
panel = tkinter.Label(top, image = img)
panel.place(y=100, x=75)


# Since every computer manages the COMport differently, it's ask
to the user on which port the XBee modules has been connected

port = input("Enter the COMPort you are using\n")
try:
    ser = serial.Serial(
        port='COM'+port,\
        baudrate=9600,\
        parity=serial.PARITY_NONE,\
        stopbits=serial.STOPBITS_ONE,\
        bytesize=serial.EIGHTBITS,\
            timeout=0)
    print("connected to: " + ser.portstr)
# If the port introduced by the user is not accessible a Print
on the screen appears and the program shuts down
except:

    print('Cannot acces port COM'+port)
    exit()

ser.flushInput()
ser.flushOutput()

data_raw_t=[0]
data_t = []
data2_t =[]
```

```
data_raw_v=[0]
data_v = []
data2_v =[]
data_raw_c=[0]
data_c = []
data2_c =[]
size=(4,3)
num=0

fig= plt.figure(figsize=size)
fig.autofmt_xdate()
fig, (ax1, ax2, ax3) = plt.subplots(3, sharex=True,
sharey=False)

ax1.set_ylabel('Voltage')
ax2.set_ylabel('Current')
ax3.set_ylabel('Thrust')
fig.subplots_adjust(hspace=0.1)
plt.setp([a.get_xticklabels() for a in fig.axes[:-1]],
visible=False)

counter_v=0
counter_t=0
counter_c=0

plt.xticks(rotation=50)

canvas = FigureCanvasTkAgg(fig, master=top)
canvas.get_tk_widget().place(y=-100, x=300)
canvas.get_tk_widget().config(width=600, height=850)

first.lift()

# This is the function that is call recursively to update the
graph
# This function checks the values of the checkboxes on the GUI
and passes the variables onto the functions that gather the data
from the COMport
# Also it manages which function to call in function of which
carachter besides < has been read. A V for Voltage, a T for
Thrust or a C for Current
def update(i):

    data_raw=[0]
    global counter
    global data
    global data2
    global VR2
    global VR1
```

```
    VR1=0
    VR2=0

    alarms_activated = True

    if CheckVar2.get()==1 :
        setup= True
        VR2=int(R2.get())
        VR1=int(R1.get())

    elif CheckVar2.get() == 0:
        setup= False
        first.lift()
    incomingByte = ser.read().decode('utf-8')

    if CheckVar1.get()==1:
        alarms_activated = True

    elif CheckVar1.get() == 0:
        alarms_activated = False

    if incomingByte =='<':
        func = ser.read().decode('utf-8')
        if func== 'v':
            v(incomingByte,alarms_activated,setup,VR1,VR2)
        elif func=='t':
            t(incomingByte,alarms_activated,setup)
        elif func=='c' :
            c(incomingByte,alarms_activated,setup)


if __name__ == '__main__':

    # This is the main of the program, it includes the main loop
of the GUI and the call of the funcion update, each 350ms
stating the sample time.


    anim=anim.FuncAnimation(fig, update, frames=60,
interval=350)

    top.mainloop()
```

Annex B: Software's Readme

This program was developed for a Bachelor's Degree Final Project to deliver to Venturi Unmanned Technologies, a tool to monitor the data that their bench test was gathering and sending through XBee network.

For the program to run correctly, first you need to locate on which COMport you XBee End Device is connected.

Execute the main program.

It asks for the COMport. Once you enter from the keyboard the number of the COM port you are using, the GUI will appear.

Enter the value of the resistors and select done.

Immediately the program will start to gather all the values that the XBee is receiving and will plot them in the graph.

If you want to activate the thresholds, just enter all of the values and select Alarms.

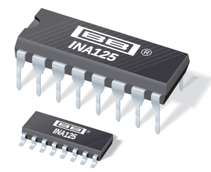
Every time that one threshold is surpassed it will send an email to the email address written on the Source code.

For the session to end just click the button, or close the window, if you want to check the values on the CSV, click the button Open CSV.

The CSV is always stored in the same directory as the program.

Annex C: Component's datasheets

BURR - BROWN®



INA125

INSTRUMENTATION AMPLIFIER With Precision Voltage Reference

FEATURES

- LOW QUIESCENT CURRENT: 460µA
- PRECISION VOLTAGE REFERENCE: 1.24V, 2.5V, 5V or 10V
- SLEEP MODE
- LOW OFFSET VOLTAGE: 250µV max
- LOW OFFSET DRIFT: 2µV/°C max
- LOW INPUT BIAS CURRENT: 20nA max
- HIGH CMR: 100dB min
- LOW NOISE: 38nV/$\sqrt{\text{Hz}}$ at f = 1kHz
- INPUT PROTECTION TO ±40V
- WIDE SUPPLY RANGE
  Single Supply: 2.7V to 36V
  Dual Supply: ±1.35V to ±18V
- 16-PIN DIP AND SO-16 SOIC PACKAGES

APPLICATIONS

- PRESSURE AND TEMPERATURE BRIDGE AMPLIFIERS
- INDUSTRIAL PROCESS CONTROL
- FACTORY AUTOMATION
- MULTI-CHANNEL DATA ACQUISITION
- BATTERY OPERATED SYSTEMS
- GENERAL PURPOSE INSTRUMENTATION

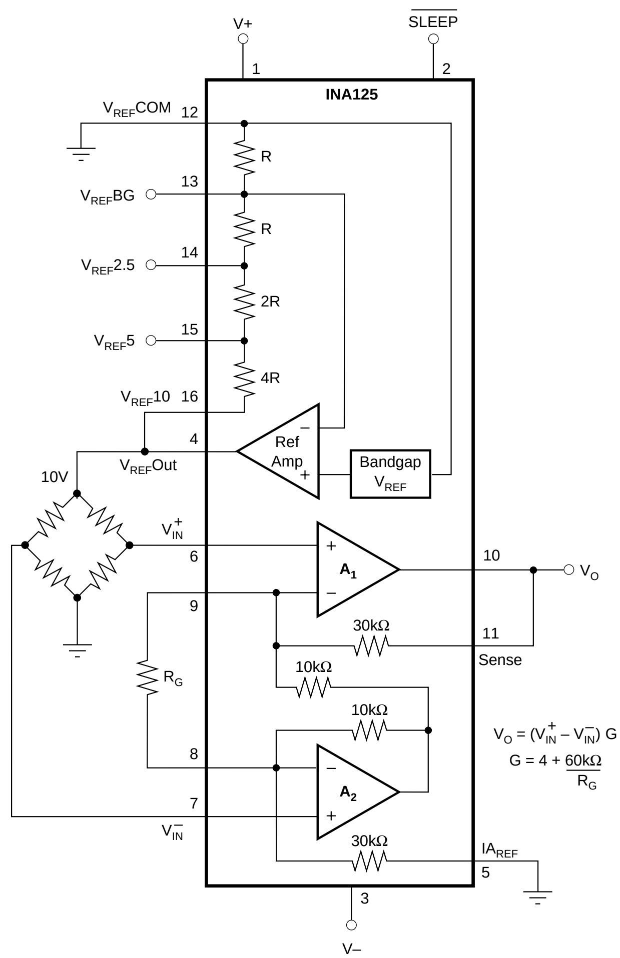
DESCRIPTION

The INA125 is a low power, high accuracy instrumentation amplifier with a precision voltage reference. It provides complete bridge excitation and precision differential-input amplification on a single integrated circuit.

A single external resistor sets any gain from 4 to 10,000. The INA125 is laser-trimmed for low offset voltage (250µV), low offset drift (2µV/°C), and high common-mode rejection (100dB at G = 100). It operates on single (+2.7V to +36V) or dual (±1.35V to ±18V) supplies.

The voltage reference is externally adjustable with pin-selectable voltages of 2.5V, 5V, or 10V, allowing use with a variety of transducers. The reference voltage is accurate to ±0.5% (max) with ±35ppm/°C drift (max). Sleep mode allows shutdown and duty cycle operation to save power.

The INA125 is available in 16-pin plastic DIP and SO-16 surface-mount packages and is specified for the –40°C to +85°C industrial temperature range.



International Airport Industrial Park • Mailing Address: PO Box 11400, Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd., Tucson, AZ 85706 • Tel: (520) 746-1111 • Twx: 910-952-1111
Internet: http://www.burr-brown.com/ • FAXLine: (800) 548-6133 (US/Canada Only) • Cable: BBRCORP • Telex: 066-6491 • FAX: (520) 889-1510 • Immediate Product Info: (800) 548-6132

 PDS-1361B Printed in U.S.A., February, 1998

SBOS060

SPECIFICATIONS: V_S = ±15V

At T_A = +25°C, V_S = ±15V, IA common = 0V, V_{REF} common = 0V, and R_L = 10kΩ, unless otherwise noted.

PARAMETER	CONDITIONS	INA125P, U MIN	INA125P, U TYP	INA125P, U MAX	INA125PA, UA MIN	INA125PA, UA TYP	INA125PA, UA MAX	UNITS
INPUT								
Offset Voltage, RTI								
Initial			±50	±250		*	±500	μV
vs Temperature			±0.25	±2		*	±5	μV/°C
vs Power Supply	V_S = ±1.35V to ±18V, G = 4		±3	±20		*	±50	μV/V
Long-Term Stability			±0.2			*		μV/mo
Impedance, Differential			$10^{11} \parallel 2$			*		Ω ∥ pF
Common-Mode			$10^{11} \parallel 9$			*		Ω ∥ pF
Safe Input Voltage				±40			*	V
Input Voltage Range			See Text			*		
Common-Mode Rejection	V_{CM} = –10.7V to +10.2V							
	G = 4	78	84		72	*		dB
	G = 10	86	94		80	*		dB
	G = 100	100	114		90	*		dB
	G = 500	100	114		90	*		dB
BIAS CURRENT	V_{CM} = 0V		10	25		*	50	nA
vs Temperature			±60			*		pA/°C
Offset Current			±0.5	±2.5		*	±5	nA
vs Temperature			±0.5			*		pA/°C
NOISE, RTI	R_S = 0Ω							
Voltage Noise, f = 10Hz			40			*		nV/√Hz
f = 100Hz			38			*		nV/√Hz
f = 1kHz			38			*		nV/√Hz
f = 0.1Hz to 10Hz			0.8			*		μVp-p
Current Noise, f = 10Hz			170			*		fA/√Hz
f = 1kHz			56			*		fA/√Hz
f = 0.1Hz to 10Hz			5			*		pAp-p
GAIN								
Gain Equation			4 + 60kΩ/R_G			*		V/V
Range of Gain		4		10,000	*		*	V/V
Gain Error	V_O = –14V to +13.3V							
	G = 4		±0.01	±0.075		*	±0.1	%
	G = 10		±0.03	±0.3		*	±0.5	%
	G = 100		±0.05	±0.5		*	±1	%
	G = 500		±0.1			*		%
Gain vs Temperature								
	G = 4		±1	±15		*	*	ppm/°C
	G > 4(1)		±25	±100		*	*	ppm/°C
Nonlinearity	V_O = –14V to +13.3V							
	G = 4		±0.0004	±0.002		*	±0.004	% of FS
	G = 10		±0.0004	±0.002		*	±0.004	% of FS
	G = 100		±0.001	±0.01		*	*	% of FS
	G = 500		±0.002			*		% of FS
OUTPUT								
Voltage: Positive		(V+)–1.7	(V+)–0.9		*	*		V
Negative		(V–)+1	(V–)+0.4		*	*		V
Load Capacitance Stability			1000			*		pF
Short-Circuit Current			–9/+12			*		mA
VOLTAGE REFERENCE	V_{REF} = +2.5V, +5V, +10V							
Accuracy	I_L = 0		±0.15	±0.5		*	±1	%
vs Temperature	I_L = 0		±18	±35		*	±100	ppm/°C
vs Power Supply, V+	V+ = (V_{REF} + 1.25V) to +36V		±20	±50		*	±100	ppm/V
vs Load	I_L = 0 to 5mA		3	75		*	*	ppm/mA
Dropout Voltage, (V+) – V_{REF}(2)	Ref Load = 2kΩ	1.25	1		*	*		V
Bandgap Voltage Reference			1.24			*		V
Accuracy	I_L = 0		±0.5			*		%
vs Temperature	I_L = 0		±18			*		ppm/°C

SPECIFICATIONS: V_S = ±15V (CONT)

At T_A = +25°C, V_S = ±15V, IA common = 0V, V_{REF} common = 0V, and R_L = 10kΩ, unless otherwise noted.

PARAMETER CONDITIONS		INA125P, U			INA125PA, UA			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
FREQUENCY RESPONSE								
Bandwidth, –3dB	G = 4		150			∗		kHz
	G = 10		45			∗		kHz
	G = 100		4.5			∗		kHz
	G = 500		0.9			∗		kHz
Slew Rate	G = 4, 10V Step		0.2			∗		V/µs
Settling Time, 0.01%	G = 4, 10V Step		60			∗		µs
	G = 10, 10V Step		83			∗		µs
	G = 100, 10V Step		375			∗		µs
	G = 500, 10V Step		1700			∗		µs
Overload Recovery	50% Overdrive		5			∗		µs
POWER SUPPLY								
Specified Operating Voltage			±15			∗		V
Specified Voltage Range		±1.35		±18	∗		∗	V
Quiescent Current, Positive	$I_O = I_{REF}$ = 0mA		460	525		∗	∗	µA
Negative	$I_O = I_{REF}$ = 0mA		–280	–325		∗	∗	µA
Reference Ground Current(3)			180			∗		µA
Sleep Current (V_{SLEEP} ≤ 100mV)	R_L = 10kΩ, Ref Load = 2kΩ		±1	±25		∗	∗	µA
SLEEP MODE PIN(4)								
V_{IH} (Logic high input voltage)		+2.7		V+	∗		∗	V
V_{IL} (Logic low input voltage)		0		+0.1	∗		∗	V
I_{IH} (Logic high input current)			15			∗		µA
I_{IL} (Logic low input current)			0			∗		µA
Wake-up Time(5)			150			∗		µs
TEMPERATURE RANGE								
Specification Range		–40		+85	∗		∗	°C
Operation Range		–55		+125	∗		∗	°C
Storage Range		–55		+125	∗		∗	°C
Thermal Resistance, θ_{JA}								
16-Pin DIP			80			∗		°C/W
SO-16 Surface-Mount			100			∗		°C/W

∗ Specification same as INA125P, U.

NOTES: (1) Temperature coefficient of the "Internal Resistor" in the gain equation. Does not include TCR of gain-setting resistor, R_G. (2) Dropout voltage is the positive supply voltage minus the reference voltage that produces a 1% decrease in reference voltage. (3) V_{REF}COM pin. (4) Voltage measured with respect to Reference Common. Logic low input selects Sleep mode. (5) IA and Reference, see Typical Performance Curves.

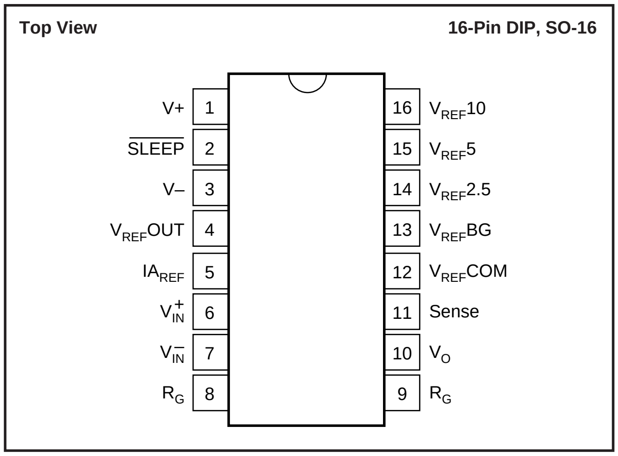
SPECIFICATIONS: V_S = +5V

At T_A = +25°C, V_S = +5V, IA common at V_S/2, V_{REF} common = V_S/2, V_{CM} = V_S/2, and R_L = 10kΩ to V_S/2, unless otherwise noted.

PARAMETER	CONDITIONS	INA125P, U			INA125PA, UA			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
INPUT								
Offset Voltage, RTI								
Initial			±75	±500		∗	±750	µV
vs Temperature			±0.25			∗		µV/°C
vs Power Supply	V_S = +2.7V to +36V		3	20		∗	50	µV/V
Input Voltage Range			See Text			∗		
Common-Mode Rejection	V_{CM} = +1.1V to +3.6V							
	G = 4	78	84		72	∗		dB
	G = 10	86	94		80	∗		dB
	G = 100	100	114		90	∗		dB
	G = 500	100	114		90	∗		dB
GAIN								
Gain Error	V_O = +0.3V to +3.8V							
	G = 4		±0.01			∗		%
OUTPUT								
Voltage, Positive		(V+)–1.2	(V+)–0.8		∗	∗		V
Negative		(V–)+0.3	(V–)+0.15		∗	∗		V
POWER SUPPLY								
Specified Operating Voltage			+5			∗		V
Operating Voltage Range		+2.7		+36	∗		∗	V
Quiescent Current	$I_O = I_{REF}$ = 0mA		460	525		∗	∗	µA
Sleep Current (V_{SLEEP} ≤ 100mV)	R_L = 10kΩ, Ref Load = 2kΩ		±1	±25		∗	∗	µA

∗ Specification same as INA125P, U.

PIN CONFIGURATION



ELECTROSTATIC DISCHARGE SENSITIVITY

This integrated circuit can be damaged by ESD. Burr-Brown recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ABSOLUTE MAXIMUM RATINGS(1)

Power Supply Voltage, V+ to V–	36V
Input Signal Voltage	±40V
Output Short Circuit	Continuous
Operating Temperature	–55°C to +125°C
Storage Temperature	–55°C to +125°C
Lead Temperature (soldering, 10s)	+300°C

NOTE: Stresses above these ratings may cause permanent damage.

PACKAGE INFORMATION

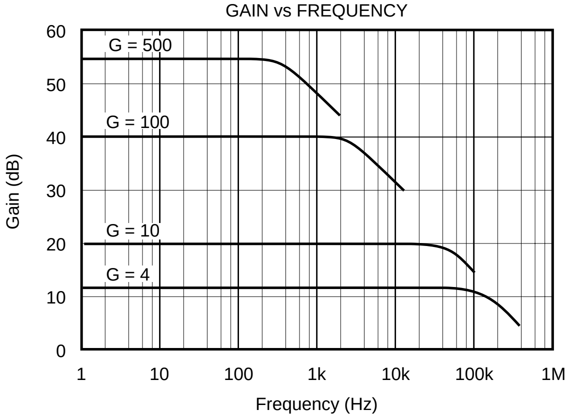
PRODUCT	PACKAGE	PACKAGE DRAWING NUMBER(1)
INA125PA	16-Pin Plastic DIP	180
INA125P	16-Pin Plastic DIP	180
INA125UA	SO-16 Surface-Mount	265
INA125U	SO-16 Surface-Mount	265

NOTES: (1) For detailed drawing and dimension table, please see end of data sheet, or Appendix C of Burr-Brown IC Data Book.

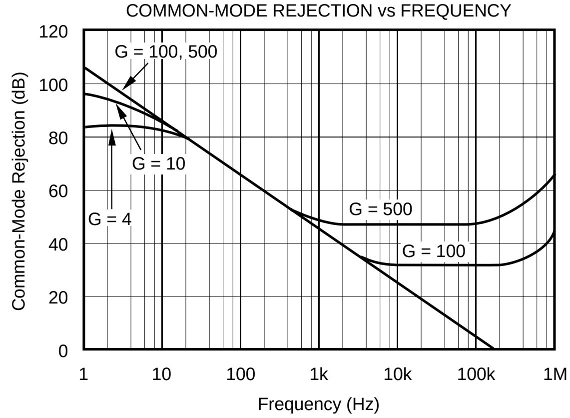
TYPICAL PERFORMANCE CURVES

At T_A = +25°C and V_S = ±15V, unless otherwise noted.

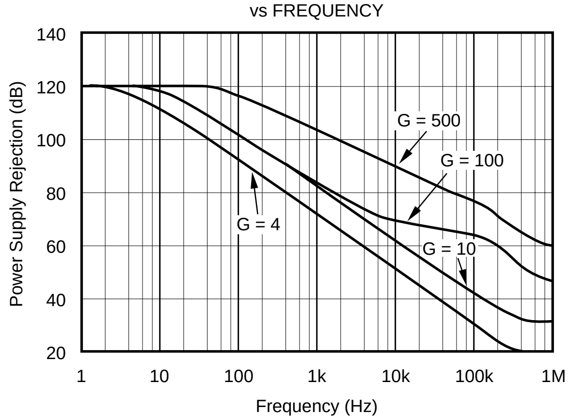
GAIN vs FREQUENCY



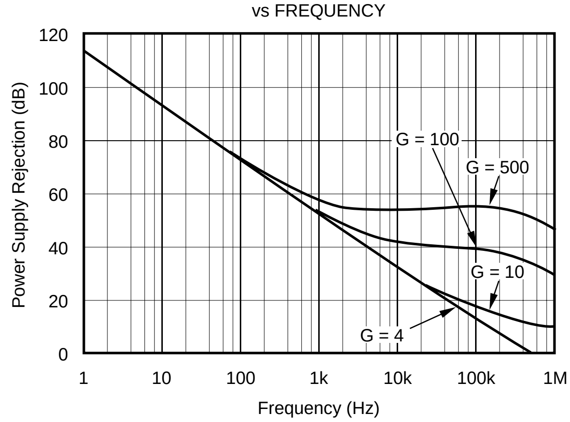
COMMON-MODE REJECTION vs FREQUENCY



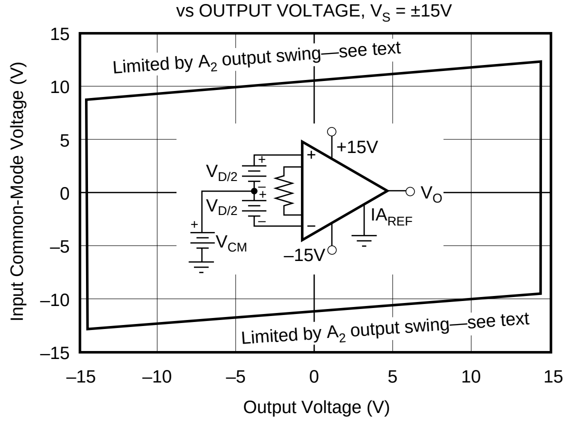
POSITIVE POWER SUPPLY REJECTION
vs FREQUENCY



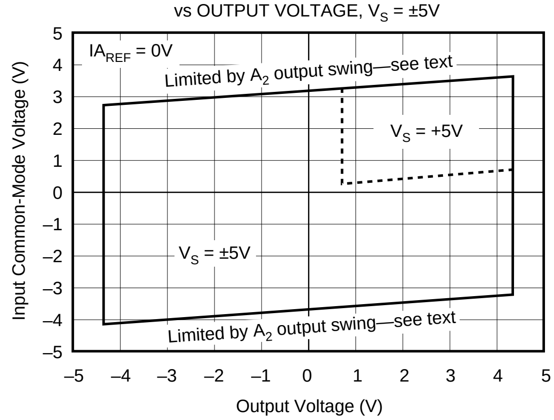
NEGATIVE POWER SUPPLY REJECTION
vs FREQUENCY



INPUT COMMON-MODE VOLTAGE
vs OUTPUT VOLTAGE, V_S = ±15V

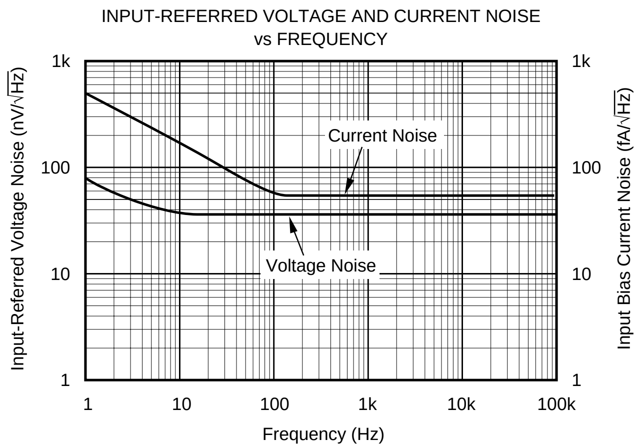


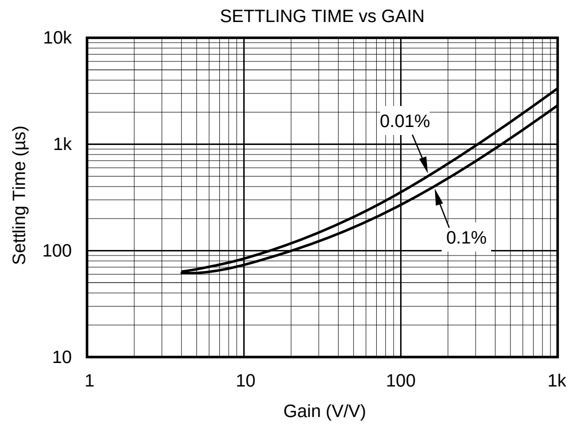
INPUT COMMON-MODE VOLTAGE
vs OUTPUT VOLTAGE, V_S = ±5V

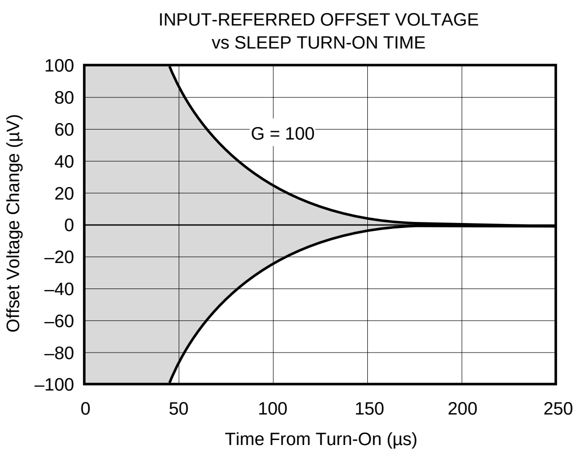


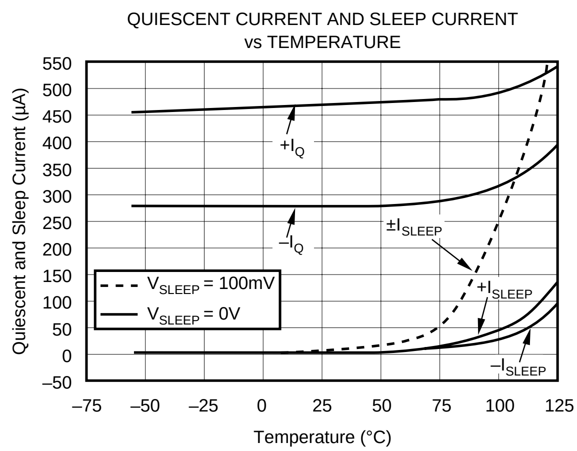
TYPICAL PERFORMANCE CURVES (CONT)

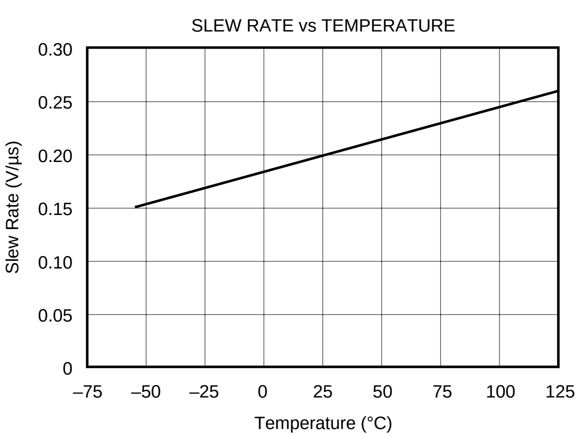
At T_A = +25°C and V_S = ±15V, unless otherwise noted.

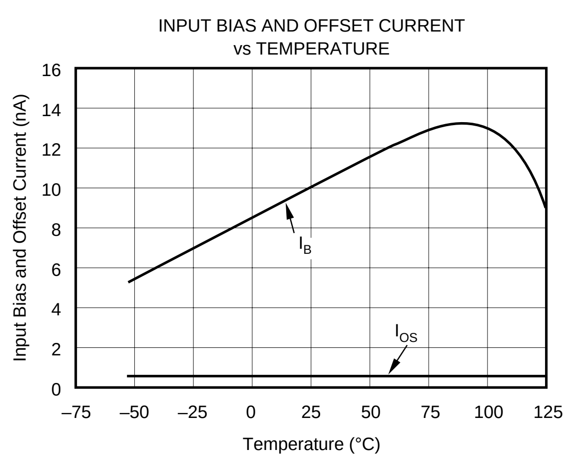








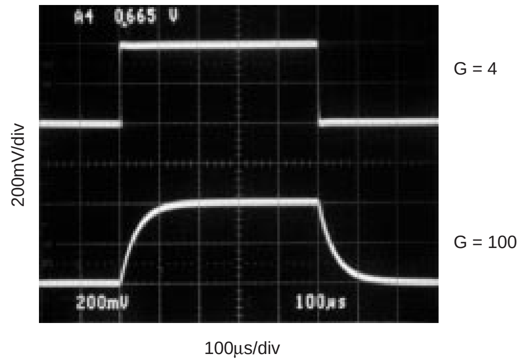




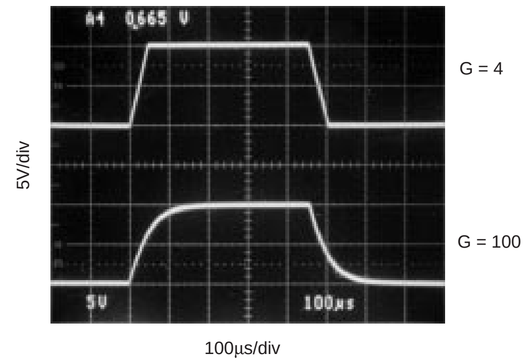
TYPICAL PERFORMANCE CURVES (CONT)

At T_A = +25°C and V_S = ±15V, unless otherwise noted.

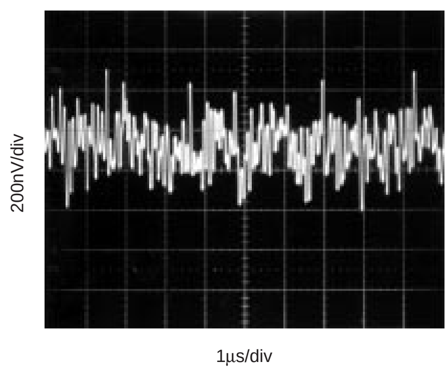
SMALL-SIGNAL RESPONSE



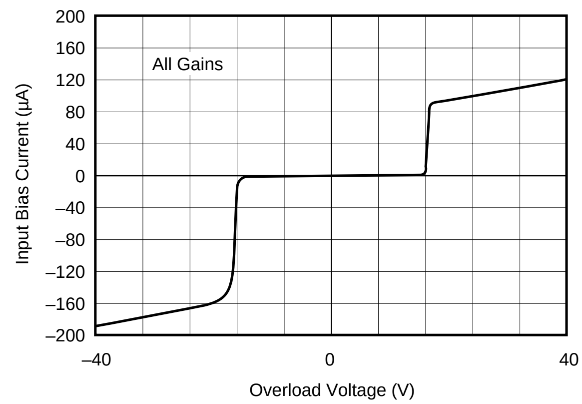
LARGE-SIGNAL RESPONSE



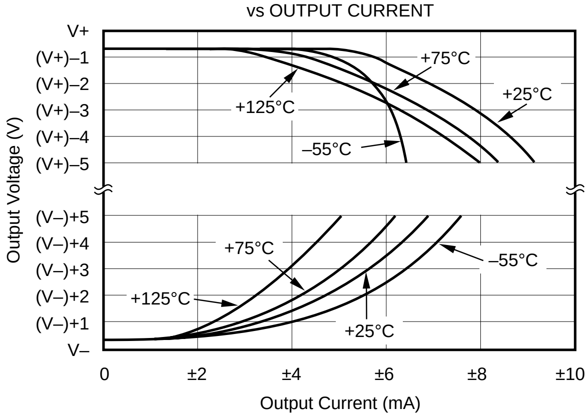
INPUT-REFERRED NOISE, 0.1Hz to 10Hz



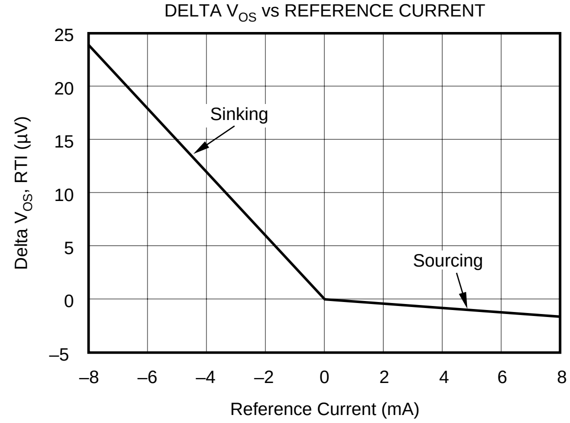
INPUT BIAS CURRENT vs INPUT OVERLOAD VOLTAGE



OUTPUT VOLTAGE SWING vs OUTPUT CURRENT



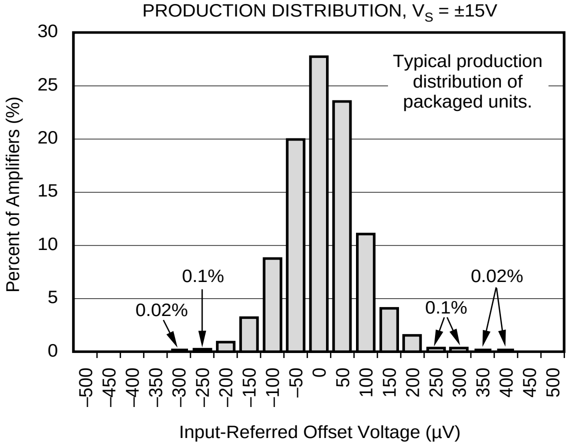
DELTA V_{OS} vs REFERENCE CURRENT



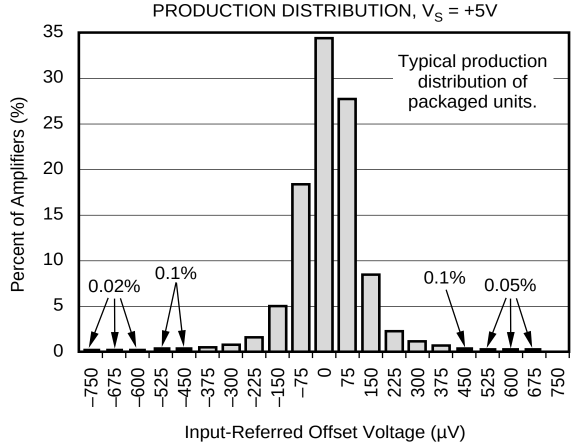
TYPICAL PERFORMANCE CURVES (CONT)

At T_A = +25°C and V_S = ±15V, unless otherwise noted.

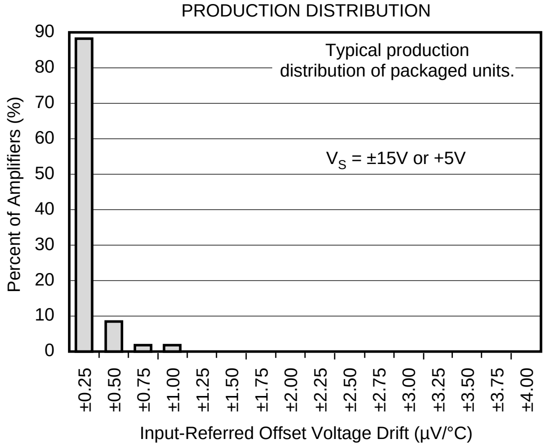
INPUT-REFERRED OFFSET VOLTAGE PRODUCTION DISTRIBUTION, V_S = ±15V



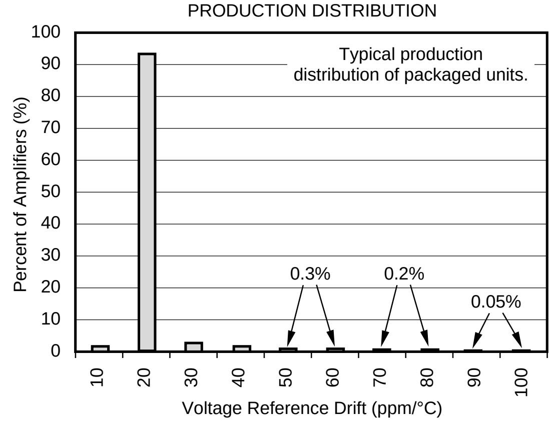
INPUT-REFERRED OFFSET VOLTAGE PRODUCTION DISTRIBUTION, V_S = +5V



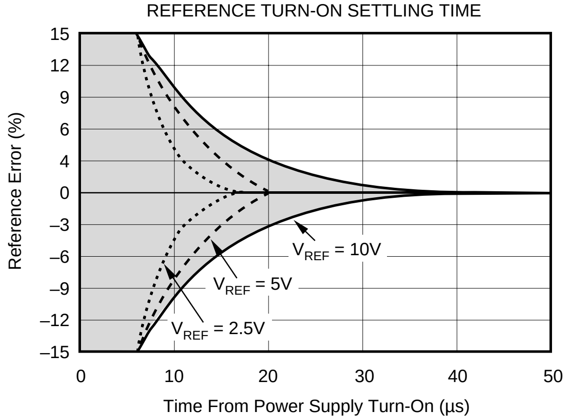
INPUT-REFERRED OFFSET VOLTAGE DRIFT PRODUCTION DISTRIBUTION



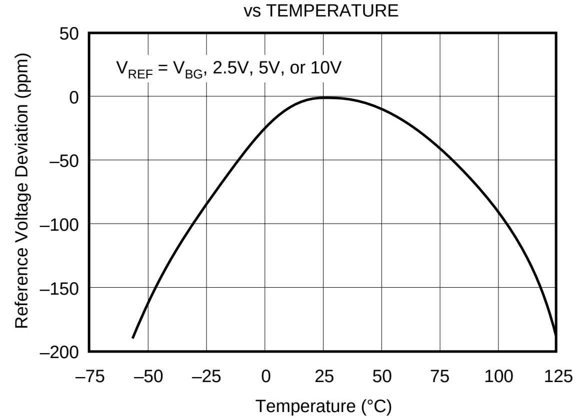
VOLTAGE REFERENCE DRIFT PRODUCTION DISTRIBUTION



REFERENCE TURN-ON SETTLING TIME



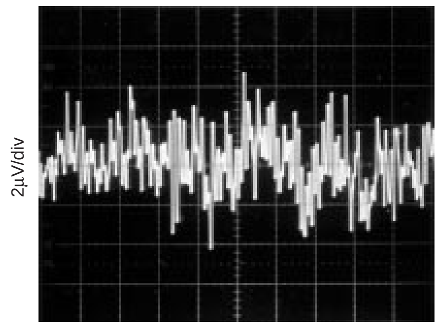
REFERENCE VOLTAGE DEVIATION vs TEMPERATURE



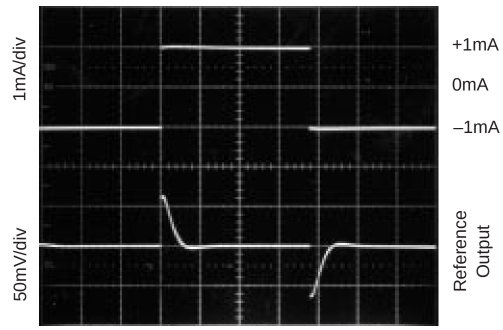
TYPICAL PERFORMANCE CURVES (CONT)

At T_A = +25°C and V_S = ±15V, unless otherwise noted.

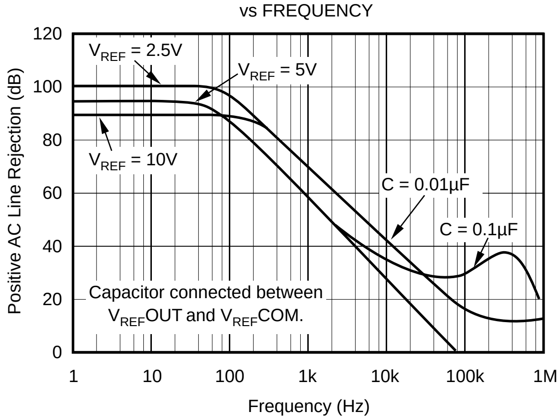
0.1Hz to 10Hz REFERENCE NOISE
V_{REF} = 2.5V, C_L = 100pF



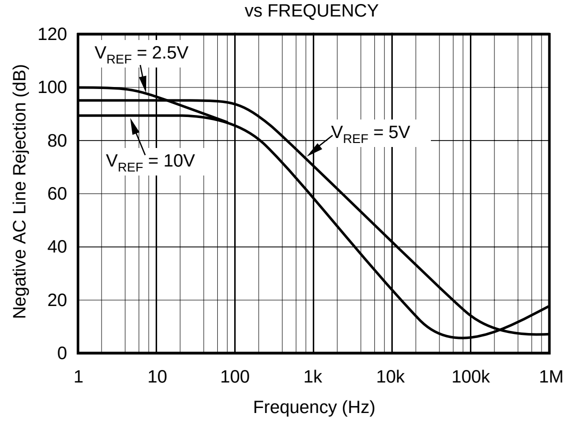
REFERENCE TRANSIENT RESPONSE
V_{REF} = 2.5V, C_L = 100pF



POSITIVE REFERENCE AC LINE REJECTION
vs FREQUENCY



NEGATIVE REFERENCE AC LINE REJECTION
vs FREQUENCY



APPLICATION INFORMATION

Figure 1 shows the basic connections required for operation of the INA125. Applications with noisy or high impedance power supplies may require decoupling capacitors close to the device pins as shown.

The output is referred to the instrumentation amplifier reference (IA_{REF}) terminal which is normally grounded. This must be a low impedance connection to assure good common-mode rejection. A resistance of 12Ω in series with the IA_{REF} pin will cause a typical device to degrade to approximately 80dB CMR (G = 4).

Connecting V_{REF}OUT (pin 4) to one of the four available reference voltage pins (V_{REF}BG, V_{REF}2.5, V_{REF}5, or V_{REF}10) provides an accurate voltage source for bridge applications. For example, in Figure 1 V_{REF}OUT is connected to V_{REF}10 thus supplying 10V to the bridge. It is recommended that V_{REF}OUT be connected to one of the reference voltage pins even when the reference is not being utilized to avoid saturating the reference amplifier. Driving the $\overline{\text{SLEEP}}$ pin LOW puts the INA125 in a shutdown mode.

SETTING THE GAIN

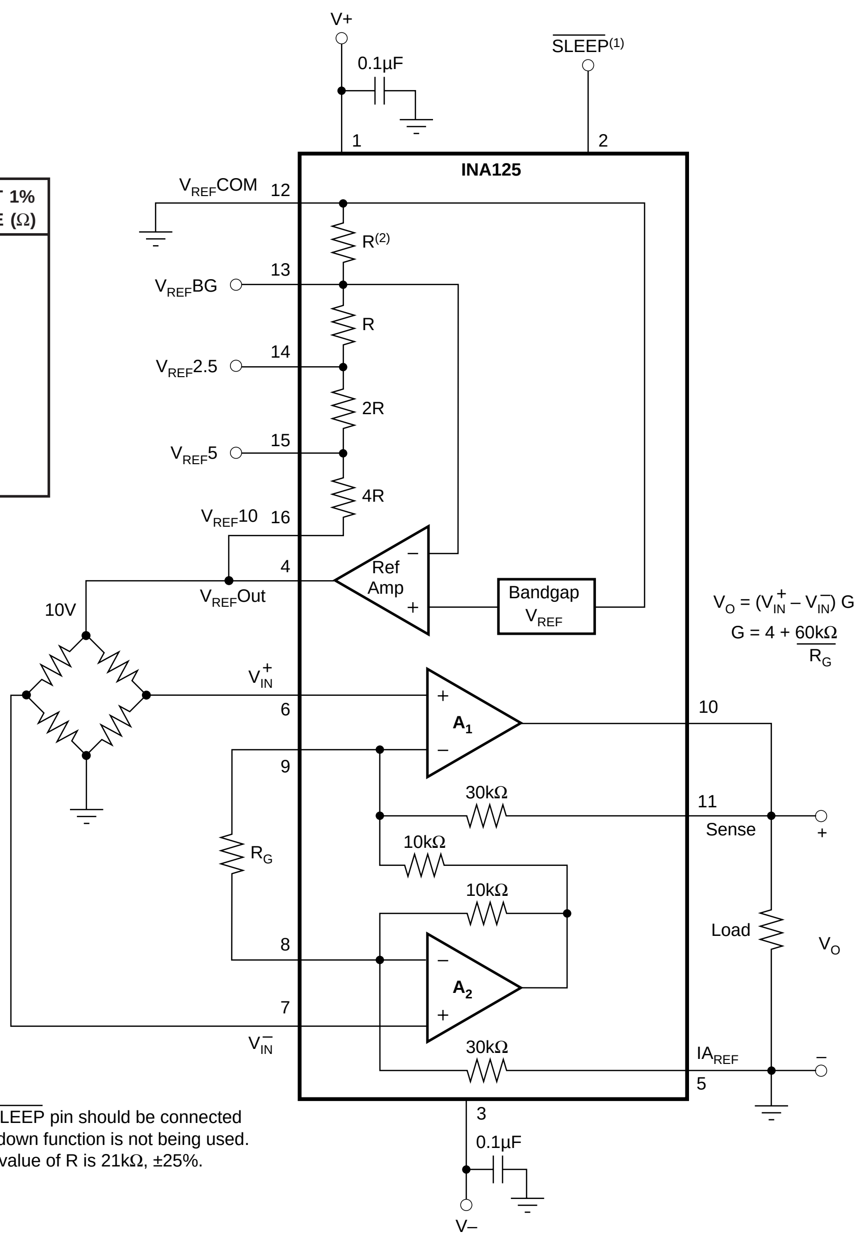
Gain of the INA125 is set by connecting a single external resistor, R_G, between pins 8 and 9:

$$G = 4 + \frac{60k\Omega}{R_G} \tag{1}$$

Commonly used gains and R_G resistor values are shown in Figure 1.

DESIRED GAIN (V/V)	R_G (Ω)	NEAREST 1% R_G VALUE (Ω)
4	NC	NC
5	60k	60.4k
10	10k	10k
20	3750	3740
50	1304	1300
100	625	619
200	306	309
500	121	121
1000	60	60.4
2000	30	30.1
10000	6	6.04

NC: No Connection.



NOTE: (1) $\overline{\text{SLEEP}}$ pin should be connected to V+ if shutdown function is not being used. (2) Nominal value of R is 21kΩ, ±25%.

FIGURE 1. Basic Connections.

The $60k\Omega$ term in equation 1 comes from the internal metal film resistors which are laser trimmed to accurate absolute values. The accuracy and temperature coefficient of these resistors are included in the gain accuracy and drift specifications of the INA125.

The stability and temperature drift of the external gain setting resistor, R_G, also affects gain. R_G's contribution to gain accuracy and drift can be directly inferred from the gain equation (1). Low resistor values required for high gain can make wiring resistance important. Sockets add to the wiring resistance, which will contribute additional gain error (possibly an unstable gain error) in gains of approximately 100 or greater.

OFFSET TRIMMING

The INA125 is laser trimmed for low offset voltage and offset voltage drift. Most applications require no external offset adjustment. Figure 2 shows an optional circuit for trimming the output offset voltage. The voltage applied to the IA_{REF} terminal is added to the output signal. The op amp buffer is used to provide low impedance at the IA_{REF} terminal to preserve good common-mode rejection.

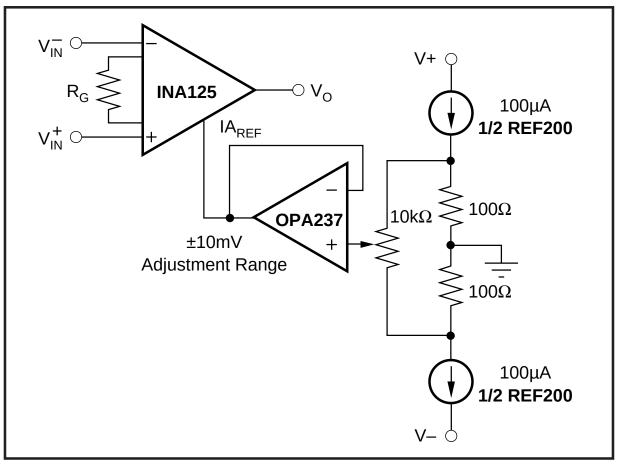


FIGURE 2. Optional Trimming of Output Offset Voltage.

INPUT BIAS CURRENT RETURN

The input impedance of the INA125 is extremely high—approximately $10^{11}\Omega$. However, a path must be provided for the input bias current of both inputs. This input bias current flows out of the device and is approximately 10nA. High input impedance means that this input bias current changes very little with varying input voltage.

Input circuitry must provide a path for this input bias current for proper operation. Figure 3 shows various provisions for an input bias current path. Without a bias current path, the inputs will float to a potential which exceeds the common-mode range, and the input amplifiers will saturate.

If the differential source resistance is low, the bias current return path can be connected to one input (see the thermocouple example in Figure 3). With higher source impedance, using two equal resistors provides a balanced input with possible advantages of lower input offset voltage due to bias current and better high frequency common-mode rejection.

INPUT COMMON-MODE RANGE

The input common-mode range of the INA125 is shown in the typical performance curves. The common-mode range is limited on the negative side by the output voltage swing of A_2, an internal circuit node that cannot be measured on an external pin. The output voltage of A2 can be expressed as:

$$V_{02} = 1.3V_{IN}^- - (V_{IN}^+ - V_{IN}^-)\ (10k\Omega/R_G)$$

(voltages referred to IA_{REF} terminal, pin 5)

The internal op amp A_2 is identical to A_1. Its output swing is limited to approximately 0.8V from the positive supply and 0.25V from the negative supply. When the input common-mode range is exceeded (A_2's output is saturated), A_1 can still be in linear operation, responding to changes in the non-inverting input voltage. The output voltage, however, will be invalid.

PRECISION VOLTAGE REFERENCE

The on-board precision voltage reference provides an accurate voltage source for bridge and other transducer applications or ratiometric conversion with analog-to-digital converters. A reference output of 2.5V, 5V or 10V is available by connecting V_{REF}OUT (pin 4) to one of the V_{REF} pins (V_{REF}2.5, V_{REF}5, or V_{REF}10). Reference voltages are laser-trimmed for low inital error and low temperature drift. Connecting V_{REF}OUT to V_{REF}BG (pin 13) produces the bandgap reference voltage (1.24V ±0.5%) at the reference output.

Positive supply voltage must be 1.25V above the desired reference voltage. For example, with V+ = 2.7V, only the 1.24V reference (V_{REF}BG) can be used. If using dual supplies V_{REF}COM can be connected to V–, increasing the

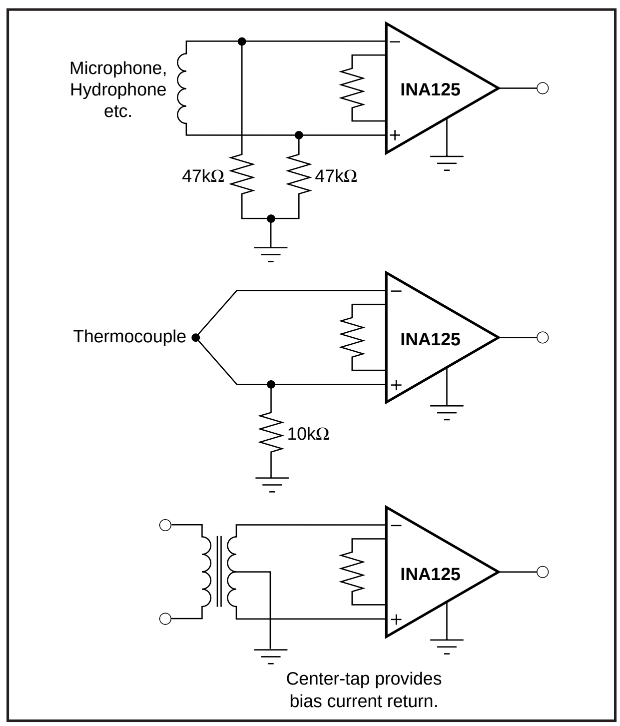


FIGURE 3. Providing an Input Common-Mode Current Path.

amount of supply voltage headroom available to the reference. Approximately 180µA flows out of the $V_{REF}COM$ terminal, therefore, it is recommended that it be connected through a low impedance path to sensor common to avoid possible ground loop problems.

Reference noise is proportional to the reference voltage selected. With V_{REF} = 2.5V, 0.1Hz to 10Hz peak-to-peak noise is approximately 9µVp-p. Noise increases to 36µVp-p for the 10V reference. Output drive capability of the voltage reference is improved by connecting a transistor as shown in Figure 4. The external transistor also serves to remove power from the INA125.

Internal resistors that set the voltage reference output are ratio-trimmed for accurate output voltages (±0.5% max). The absolute resistance values, however, may vary ±25%. Adjustment of the reference output voltage with an external resistor is not recommended because the required resistor value is uncertain.

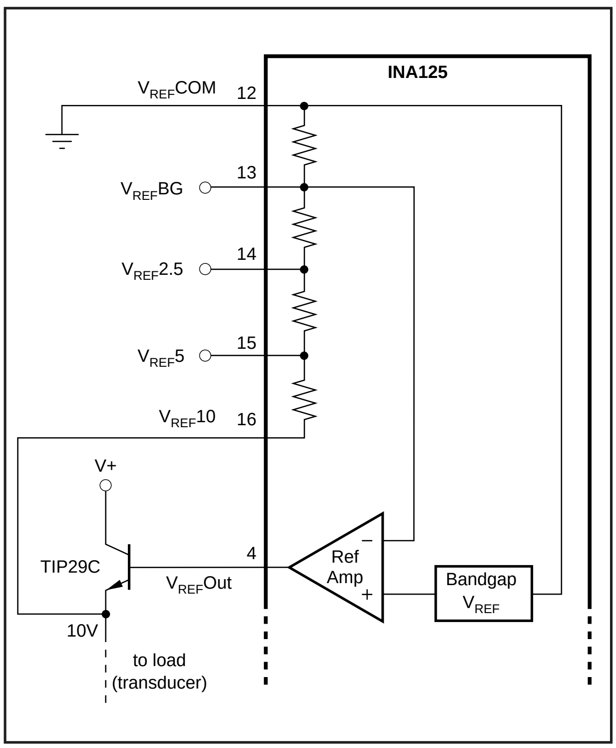


FIGURE 4. Reference Current Boost.

SHUTDOWN

The INA125 has a shutdown option. When the $\overline{\text{SLEEP}}$ pin is LOW (100mV or less), the supply current drops to approximately 1µA and output impedance becomes approximately 80kΩ. Best performance is achieved with CMOS logic. To maintain low sleep current at high temperatures, V_{SLEEP} should be as close to 0V as possible. This should not be a problem if using CMOS logic unless the CMOS gate is driving other currents. Refer to the typical performance curve, "Sleep Current vs Temperature."

A transition region exists when V_{SLEEP} is between 400mV and 2.7V (with respect to $V_{REF}COM$) where the output is unpredictable. Operation in this region is not recommended. The INA125 achieves high accuracy quickly following wake-up ($V_{SLEEP} \geq 2.7V$). See the typical performance curve "Input-Referred Offset Voltage vs Sleep Turn-on Time." If shutdown is not being used, connect the $\overline{\text{SLEEP}}$ pin to V+.

LOW VOLTAGE OPERATION

The INA125 can be operated on power supplies as low as ±1.35V. Performance remains excellent with power supplies ranging from ±1.35V to ±18V. Most parameters vary only slightly throughout this supply voltage range—see typical performance curves. Operation at very low supply voltage requires careful attention to ensure that the common-mode voltage remains within its linear range. See "Input Common-Mode Voltage Range." As previously mentioned, when using the on-board reference with low supply voltages, it may be necessary to connect $V_{REF}COM$ to V– to ensure $V_S - V_{REF} \geq 1.25V$.

SINGLE SUPPLY OPERATION

The INA125 can be used on single power supplies of +2.7V to +36V. Figure 5 shows a basic single supply circuit. The IA_{REF}, $V_{REF}COM$, and V– terminals are connected to ground. Zero differential input voltage will demand an output voltage of 0V (ground). When the load is referred to ground as shown, actual output voltage swing is limited to approximately 150mV above ground. The typical performance curve "Output Voltage Swing vs Output Current" shows how the output swing varies with output current.

With single supply operation, careful attention should be paid to input common-mode range, output voltage swing of both op amps, and the voltage applied to the IA_{REF} terminal. V_{IN+} and V_{IN-} must both be 1V above ground for linear operation. You cannot, for instance, connect the inverting input to ground and measure a voltage connected to the non-inverting input.

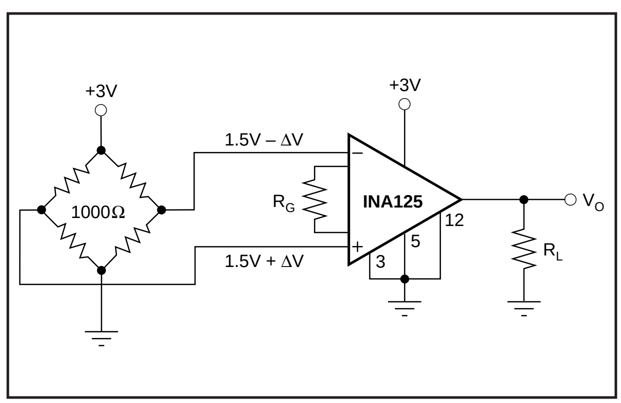


FIGURE 5. Single Supply Bridge Amplifier.

INPUT PROTECTION

The inputs of the INA125 are individually protected for voltage up to ±40V. For example, a condition of –40V on one input and +40V on the other input will not cause damage. Internal circuitry on each input provides low series impedance under normal signal conditions. To provide equivalent protection, series input resistors would contribute excessive noise. If the input is overloaded, the protection circuitry limits the input current to a safe value of approximately 120µA to 190µA. The typical performance curve "Input Bias Current vs Input Overload Voltage" shows this input current limit behavior. The inputs are protected even if the power supplies are disconnected or turned off.

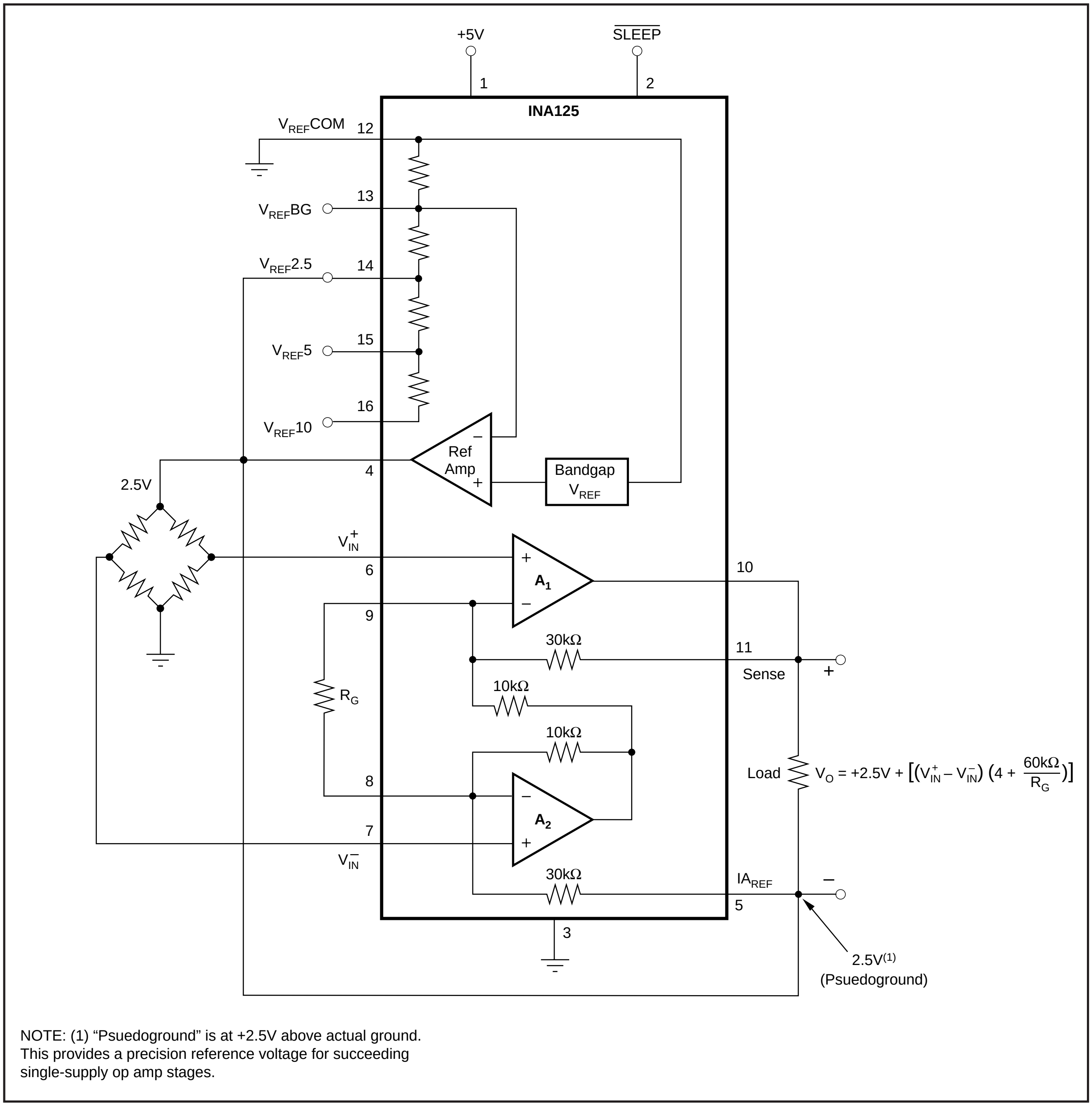


FIGURE 6. Psuedoground Bridge Measurement, 5V Single Supply.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)
INA125P	ACTIVE	PDIP	N	16	25	Green (RoHS & no Sb/Br)	CU NIPDAU	N / A for Pkg Type
INA125PA	ACTIVE	PDIP	N	16	25	Green (RoHS & no Sb/Br)	CU NIPDAU	N / A for Pkg Type
INA125PAG4	ACTIVE	PDIP	N	16	25	Green (RoHS & no Sb/Br)	CU NIPDAU	N / A for Pkg Type
INA125PG4	ACTIVE	PDIP	N	16	25	Green (RoHS & no Sb/Br)	CU NIPDAU	N / A for Pkg Type
INA125U	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR
INA125U/2K5	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR
INA125U/2K5E4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR
INA125UA	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR
INA125UA/2K5	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR
INA125UA/2K5E4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR
INA125UAG4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR
INA125UE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-3-260C-168 HR

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

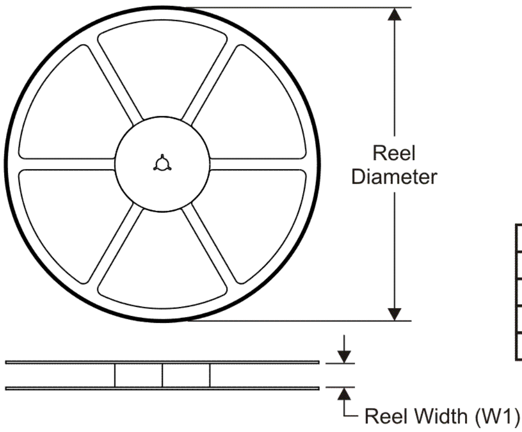
(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

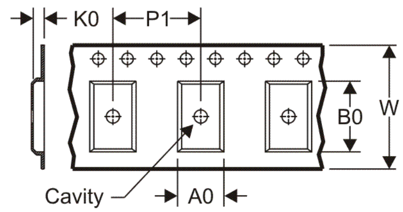
(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

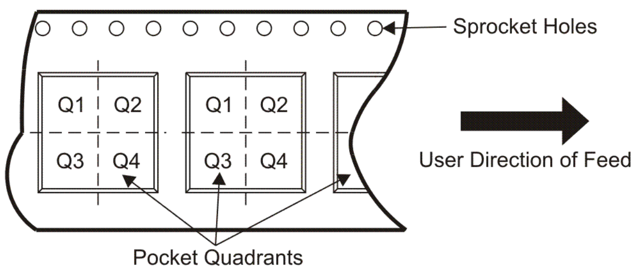
TAPE AND REEL INFORMATION





A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

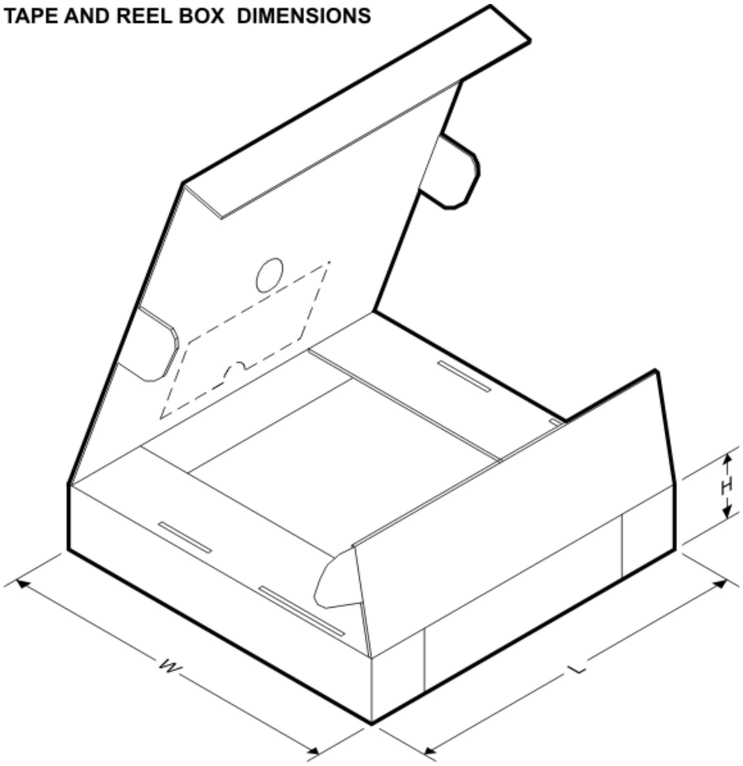
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
INA125U/2K5	SOIC	D	16	2500	330.0	16.4	6.5	10.3	2.1	8.0	16.0	Q1
INA125UA/2K5	SOIC	D	16	2500	330.0	16.4	6.5	10.3	2.1	8.0	16.0	Q1

TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
INA125U/2K5	SOIC	D	16	2500	346.0	346.0	33.0
INA125UA/2K5	SOIC	D	16	2500	346.0	346.0	33.0

TL08xx JFET-Input Operational Amplifiers

1 Features

- Low Power Consumption: 1.4 mA/ch Typical
- Wide Common-Mode and Differential Voltage Ranges
- Low Input Bias Current: 30 pA Typical
- Low Input Offset Current: 5 pA Typical
- Output Short-Circuit Protection
- Low Total Harmonic Distortion: 0.003% Typical
- High Input Impedance: JFET Input Stage
- Latch-Up-Free Operation
- High Slew Rate: 13 V/μs Typical
- Common-Mode Input Voltage Range Includes V_{CC+}

2 Applications

- Tablets
- White goods
- Personal electronics
- Computers

3 Description

The TL08xx JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset-voltage temperature coefficient.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
TL084xD	SOIC (14)	8.65 mm × 3.91 mm
TL08xxFK	LCCC (20)	8.89 mm × 8.89 mm
TL084xJ	CDIP (14)	19.56 mm × 6.92 mm
TL084xN	PDIP (14)	19.3 mm × 6.35 mm
TL084xNS	SO (14)	10.3 mm × 5.3 mm
TL084xPW	TSSOP (14)	5.0 mm × 4.4 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Schematic Symbol

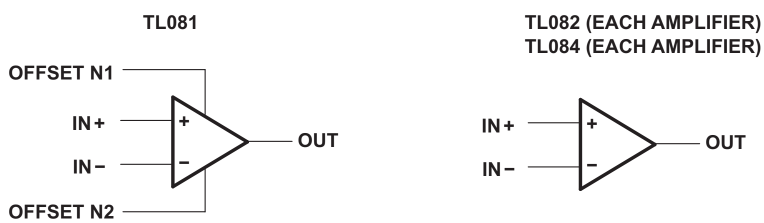


Table of Contents

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision H (January 2014) to Revision I — **Page**

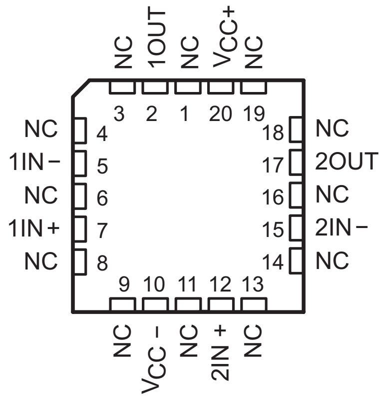
- Added *Pin Configuration and Functions* section, *Storage Conditions* table, *ESD Ratings* table, *Feature Description* section, *Device Functional Modes*, *Application and Implementation* section, *Power Supply Recommendations* section, *Layout* section, *Device and Documentation Support* section, and *Mechanical, Packaging, and Orderable Information* section ... 1
- Added *Applications* ... 1
- Moved *Typical Characteristics* into *Specifications* section. ... 9

Changes from Revision G (September 2004) to Revision H — **Page**

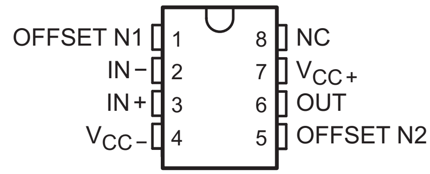
- Updated document to new TI data sheet format - no specification changes. ... 1
- Deleted *Ordering Information* table. ... 1

5 Pin Configuration and Functions

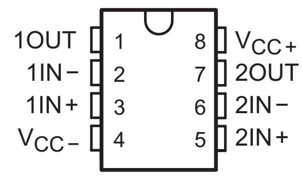
TL082 FK Package
20-Pin LCCC
Top View



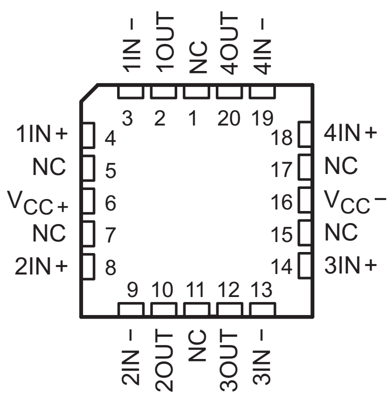
TL081 and TL081x D, P, and PS Package
8-Pin SOIC, PDIP, and SO
Top View



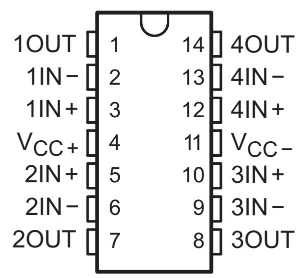
TL082 and TL082x D, JG, P, PS and PW Package
8-Pin SOIC, CDIP, PDIP, SO, and TSSOP
Top View



TL084 FK Package
20-Pin LCCC
Top View



TL084 and TL084x D, J, N, NS and PW Package
14-Pin SOIC, CDIP, PDIP, SO, and TSSOP
Top View



Pin Functions

PIN						I/O	DESCRIPTION
NAME	TL081	TL082		TL084			
	SOIC, PDIP, SO	SOIC, CDIP, PDIP, SO, TSSOP	LCCC	SOIC, CDIP, PDIP, SO, TSSOP	LCCC		
1IN–	—	2	5	2	3	I	Negative input
1IN+	—	3	7	3	4	I	Positive input
1OUT	—	1	2	1	2	O	Output
2IN–	—	6	15	6	9	I	Negative input
2IN+	—	5	12	5	8	I	Positive input
2OUT	—	7	17	7	10	O	Output
3IN–	—	—	—	9	13	I	Negative input
3IN+	—	—	—	10	14	I	Positive input
3OUT	—	—	—	8	12	O	Output
4IN–	—	—	—	13	19	I	Negative input
4IN+	—	—	—	12	18	I	Positive input
4OUT	—	—	—	14	20	O	Output

Pin Functions (continued)

<table>
<tr><th colspan="6">PIN</th><th rowspan="3">I/O</th><th rowspan="3">DESCRIPTION</th></tr>
<tr><th rowspan="2">NAME</th><th>TL081</th><th colspan="2">TL082</th><th colspan="2">TL084</th></tr>
<tr><th>SOIC, PDIP, SO</th><th>SOIC, CDIP, PDIP, SO, TSSOP</th><th>LCCC</th><th>SOIC, CDIP, PDIP, SO, TSSOP</th><th>LCCC</th></tr>
<tr><td>IN–</td><td>2</td><td>—</td><td>—</td><td>—</td><td>—</td><td>I</td><td>Negative input</td></tr>
<tr><td>IN+</td><td>3</td><td>—</td><td>—</td><td>—</td><td>—</td><td>I</td><td>Positive input</td></tr>
<tr><td rowspan="11">NC</td><td rowspan="11">8</td><td rowspan="11">—</td><td>1</td><td rowspan="11">—</td><td rowspan="2">1</td><td rowspan="11">—</td><td rowspan="11">Do not connect</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td rowspan="2">5</td></tr>
<tr><td>6</td></tr>
<tr><td>8</td><td rowspan="2">7</td></tr>
<tr><td>9</td></tr>
<tr><td>11</td><td rowspan="2">11</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td><td rowspan="2">15</td></tr>
<tr><td>16</td></tr>
<tr><td>18</td><td>17</td></tr>
<tr><td>OFFSET N1</td><td>1</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>Input offset adjustment</td></tr>
<tr><td>OFFSET N2</td><td>5</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>Input offset adjustment</td></tr>
<tr><td>OUT</td><td>6</td><td>—</td><td>—</td><td>—</td><td>—</td><td>O</td><td>Output</td></tr>
<tr><td>V_{CC-}</td><td>4</td><td>4</td><td>10</td><td>11</td><td>16</td><td>—</td><td>Power supply</td></tr>
<tr><td>V_{CC+}</td><td>7</td><td>8</td><td>20</td><td>4</td><td>6</td><td>—</td><td>Power supply</td></tr>
</table>

6 Specifications

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)[1]

				MIN	MAX	UNIT
V_{CC+}	Supply voltage[2]				18	V
V_{CC-}					–18	
V_{ID}	Differential input voltage[3]				±30	V
V_I	Input voltage[2][4]				±15	V
	Duration of output short circuit[5]			Unlimited		
	Continuous total power dissipation			See *Dissipation Rating Table*		
T_A	Operating free-air temperature		TL08_C TL08_AC TL08_BC	0	70	°C
			TL08_I	–40	85	
			TL084Q	–40	125	
			TL08_M	–55	125	
	Operating virtual junction temperature				150	°C
T_C	Case temperature for 60 seconds	FK package	TL08_M		260	°C
	Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	J or JG package	TL08_M		300	°C
T_{stg}	Storage temperature			–65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) All voltage values, except differential voltages, are with respect to the midpoint between V_{CC+} and V_{CC-}.
(3) Differential voltages are at IN+, with respect to IN−.
(4) The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 V, whichever is less.
(5) The output may be shorted to ground or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.

6.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001[1]	1000	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101[2]	1500	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

			MIN	MAX	UNIT
V_{CC+}	Supply voltage		5	15	V
V_{CC-}	Supply voltage		–5	–15	V
V_{CM}	Common-mode voltage		$V_{CC-} + 4$	$V_{CC+} - 4$	V
T_A	Ambient temperature	TL08xM	–55	125	°C
		TL08xQ	–40	125	
		TL08xI	–40	85	
		TL08xC	0	70	

6.4 Thermal Information

THERMAL METRIC[(1)]		TL08xx								UNIT
		D (SOIC)		N (PDIP)	NS (SO)	P (PDIP)	PS (SO)	PW (TSSOP)		
		8 PINS	14 PINS	14 PINS	14 PINS	{PIN COUNT} PINS	{PIN COUNT} PINS	8 PINS	14 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance[(2)(3)]	97	86	76	80	85	95	149	113	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.
(2) Maximum power dissipation is a function of $T_{J(max)}$, $R_{\theta JA}$, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_{J(max)} – T_A) / R_{\theta JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
(3) The package thermal impedance is calculated in accordance with JESD 51-7.

6.5 Electrical Characteristics for TL08xC, TL08xxC, and TL08xI

$V_{CC\pm}$ = ±15 V (unless otherwise noted)

PARAMETER		TEST CONDITIONS	T_A[(1)]	TL081C, TL082C, TL084C			TL081AC, TL082AC, TL084AC			TL081BC, TL082BC, TL084BC			TL081I, TL082I, TL084I			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V_{IO}	Input offset voltage	V_O = 0, R_S = 50 Ω	25°C		3	15		3	6		2	3		3	6	mV
			Full range			20			7.5			5			9	
α_{VIO}	Temperature coefficient of input offset voltage	V_O = 0, R_S = 50 Ω	Full range		18			18			18			18		μV/°C
I_{IO}	Input offset current[(2)]	V_O = 0	25°C		5	200		5	100		5	100		5	100	pA
			Full range			2			2			2			10	nA
I_{IB}	Input bias current[(2)]	V_O = 0	25°C		30	400		30	200		30	200		30	200	pA
			Full range			10			7			7			20	nA
V_{ICR}	Common-mode input voltage range		25°C	±11	–12 to 15		±11	–12 to 15		±11	–12 to 15		±11	–12 to 15		V
V_{OM}	Maximum peak output voltage swing	R_L = 10 kΩ	25°C	±12	±13.5		±12	±13.5		±12	±13.5		±12	±13.5		V
		R_L ≥ 10 kΩ	Full range	±12			±12			±12			±12			
		R_L ≥ 2 kΩ		±10	±12		±10	±12		±10	±12		±10	±12		
A_{VD}	Large-signal differential voltage amplification	V_O = ±10 V, R_L ≥ 2 kΩ	25°C	25	200		50	200		50	200		50	200		V/mV
			Full range	15			15			25			25			
B_1	Unity-gain bandwidth		25°C		3			3			3			3		MHz
r_i	Input resistance		25°C		10^{12}			10^{12}			10^{12}			10^{12}		Ω
CMRR	Common-mode rejection ratio	$V_{IC} = V_{ICR}$min, V_O = 0, R_S = 50 Ω	25°C	70	86		75	86		75	86		75	86		dB
k_{SVR}	Supply-voltage rejection ratio ($\Delta V_{CC\pm}/\Delta V_{IO}$)	V_{CC} = ±15 V to ±9 V, V_O = 0, R_S = 50 Ω	25°C	70	86		80	86		80	86		80	86		dB

(1) All characteristics are measured under open-loop conditions with zero common-mode voltage, unless otherwise specified. Full range for T_A is 0°C to 70°C for TL08_C, TL08_AC, TL08_BC and –40°C to 85°C for TL08_I.
(2) Input bias currents of an FET-input operational amplifier are normal junction reverse currents, which are temperature sensitive, as shown in Figure 13. Pulse techniques must be used that maintain the junction temperature as close to the ambient temperature as possible.

Electrical Characteristics for TL08xC, TL08xxC, and TL08xI (continued)

$V_{CC\pm}$ = ±15 V (unless otherwise noted)

PARAMETER		TEST CONDITIONS	T_A (1)	TL081C, TL082C, TL084C			TL081AC, TL082AC, TL084AC			TL081BC, TL082BC, TL084BC			TL081I, TL082I, TL084I			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
I_{CC}	Supply current (each amplifier)	V_O = 0, No load	25°C		1.4	2.8		1.4	2.8		1.4	2.8		1.4	2.8	mA
V_{O1}/V_{O2}	Crosstalk attenuation	A_{VD} = 100	25°C		120			120			120			120		dB

6.6 Electrical Characteristics for TL08xM and TL084x

$V_{CC\pm}$ = ±15 V (unless otherwise noted)

PARAMETER		TEST CONDITIONS (1)	T_A	TL081M, TL082M			TL084Q, TL084M			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
V_{IO}	Input offset voltage	V_O = 0, R_S = 50 Ω	25°C		3	6		3	9	mV
			Full range			9			15	
α_{VIO}	Temperature coefficient of input offset voltage	V_O = 0, R_S = 50 Ω	Full range		18			18		μV/°C
I_{IO}	Input offset current (2)	V_O = 0	25°C		5	100		5	100	pA
			125°C			20			20	nA
I_{IB}	Input bias current (2)	V_O = 0	25°C		30	200		30	200	pA
			125°C			50			50	nA
V_{ICR}	Common-mode input voltage range		25°C	±11	–12 to 15		±11	–12 to 15		V
V_{OM}	Maximum peak output voltage swing	R_L = 10 kΩ	25°C	±12	±13.5		±12	±13.5		V
		R_L ≥ 10 kΩ	Full range	±12			±12			
		R_L ≥ 2 kΩ		±10	±12		±10	±12		
A_{VD}	Large-signal differential voltage amplification	V_O = ±10 V, R_L ≥ 2 kΩ	25°C	25	200		25	200		V/mV
			Full range	15			15			
B_1	Unity-gain bandwidth		25°C		3			3		MHz
r_i	Input resistance		25°C		10^{12}			10^{12}		Ω
CMRR	Common-mode rejection ratio	V_{IC} = V_{ICR}min, V_O = 0, R_S = 50 Ω	25°C	80	86		80	86		dB
k_{SVR}	Supply-voltage rejection ratio ($\Delta V_{CC\pm}/\Delta V_{IO}$)	V_{CC} = ±15 V to ±9 V, V_O = 0, R_S = 50 Ω	25°C	80	86		80	86		dB
I_{CC}	Supply current (each amplifier)	V_O = 0, No load	25°C		1.4	2.8		1.4	2.8	mA
V_{O1}/V_{O2}	Crosstalk attenuation	A_{VD} = 100	25°C		120			120		dB

(1) All characteristics are measured under open-loop conditions, with zero common-mode input voltage, unless otherwise specified.
(2) Input bias currents of a FET-input operational amplifier are normal junction reverse currents, which are temperature sensitive, as shown in Figure 13. Pulse techniques must be used that maintain the junction temperatures as close to the ambient temperature as possible.

6.7 Operating Characteristics

$V_{CC\pm}$ = ±15 V, T_A= 25°C (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Slew rate at unity gain	V_I = 10 V, R_L = 2 kΩ, C_L = 100 pF, See Figure 19	8 (1)	13		V/μs
		V_I = 10 V, R_L = 2 kΩ, C_L = 100 pF, T_A = − 55°C to 125°C, See Figure 19	5 (1)			

(1) On products compliant to MIL-PRF-38535, this parameter is not production tested.

Operating Characteristics (continued)

$V_{CC\pm}$ = ±15 V, T_A= 25°C (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_r	Rise-time	V_I = 20 V, R_L = 2 kΩ, C_L = 100 pF, See Figure 19			0.05		μs
	overshoot factor				20%		
V_n	Equivalent input noise voltage	R_S = 20 Ω	f = 1 kHz		18		nV/√Hz
			f = 10 Hz to 10 kHz		4		μV
I_n	Equivalent input noise current	R_S = 20 Ω,	f = 1 kHz		0.01		pA/√Hz
THD	Total harmonic distortion	$V_I$rms = 6 V, A_{VD} = 1, R_S ≤ 1 kΩ, R_L ≥ 2 kΩ, f = 1 kHz,			0.003%		

6.8 Dissipation Rating Table

PACKAGE	T_A ≤ 25°C POWER RATING	DERATING FACTOR	DERATE ABOVE T_A	T_A = 70°C POWER RATING	T_A = 85°C POWER RATING	T_A = 125°C POWER RATING
D (14 pin)	680 mW	7.6 mW/°C	60°C	604 m/W	490 mW	186 mW
FK	680 mW	11.0 mW/°C	88°C	680 m/W	680 mW	273 mW
J	680 mW	11.0 mW/°C	88°C	680 m/W	680 mW	273 mW
JG	680 mW	8.4 mW/°C	69°C	672 m/W	546 mW	210 mW

6.9 Typical Characteristics

Data at high and low temperatures are applicable only within the rated operating free-air temperature ranges of the various devices. The Figure numbers referenced in the following graphs are located in *Parameter Measurement Information*.

Table 1. Table of Graphs

			Figure
V_{OM}	Maximum peak output voltage	versus Frequency versus Free-air temperature versus Load resistance versus Supply voltage	Figure 1, Figure 2, Figure 3 Figure 4 Figure 5 Figure 6
A_{VD}	Large-signal differential voltage amplification	versus Free-air temperature versus Load resistance	Figure 7 Figure 8
	Differential voltage amplification	versus Frequency with feed-forward compensation	Figure 9
P_D	Total power dissipation	versus Free-air temperature	Figure 10
I_{CC}	Supply current	versus Free-air temperature versus Supply voltage	Figure 11 Figure 12
I_{IB}	Input bias current	versus Free-air temperature	Figure 13
	Large-signal pulse response	versus Time	Figure 14
V_O	Output voltage	versus Elapsed time	Figure 15
CMRR	Common-mode rejection ratio	versus Free-air temperature	Figure 16
V_n	Equivalent input noise voltage	versus Frequency	Figure 17
THD	Total harmonic distortion	versus Frequency	Figure 18

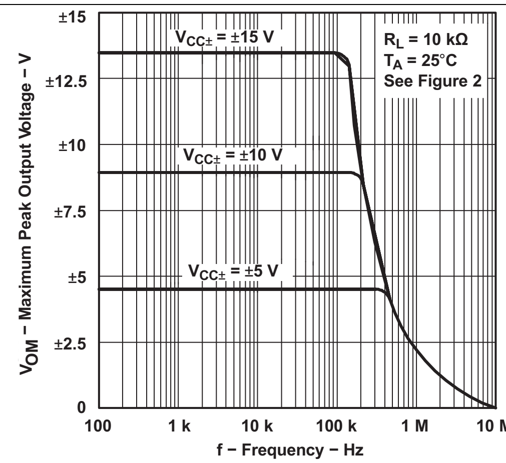


Figure 1. Maximum Peak Output Voltage vs Frequency

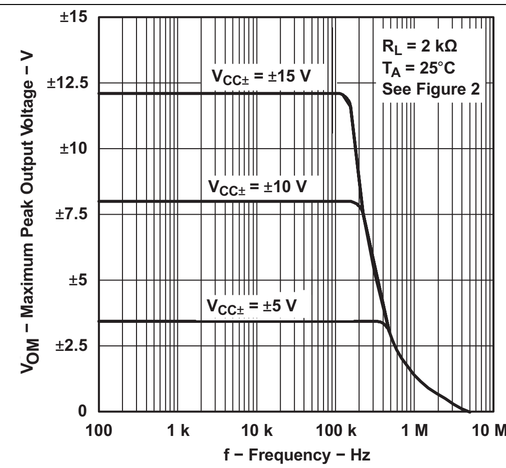


Figure 2. Maximum Peak Output Voltage vs Frequency

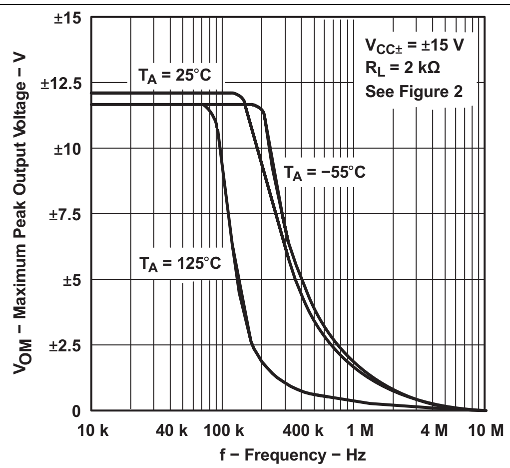


Figure 3. Maximum Peak Output Voltage vs Frequency

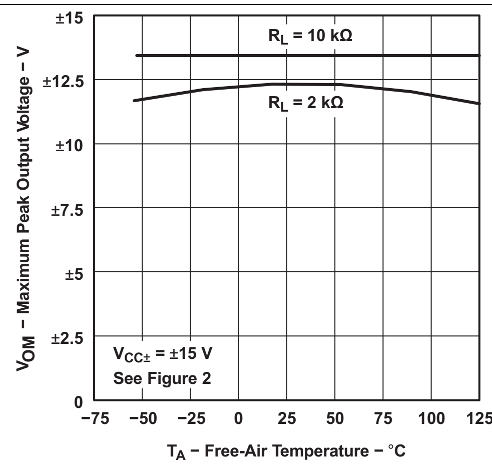


Figure 4. Maximum Peak Output Voltage vs Free-Air Temperature

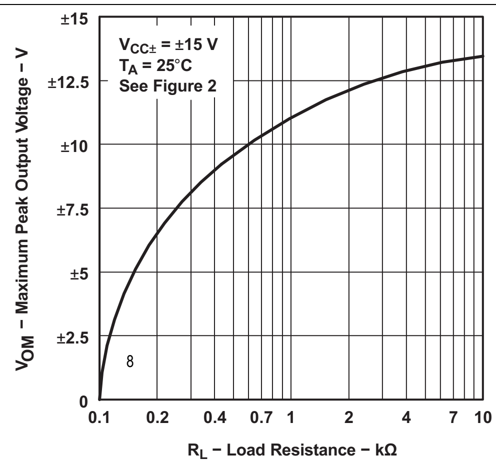


Figure 5. Maximum Peak Output Voltage vs Load Resistance

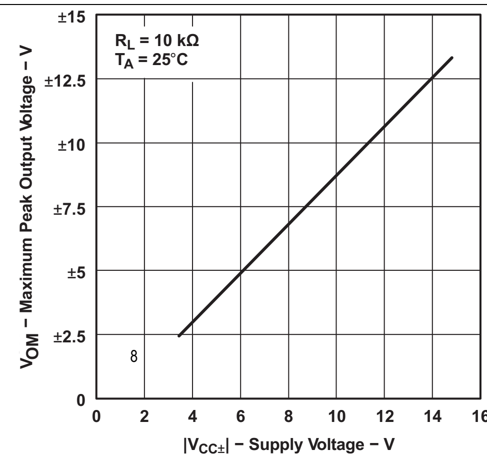


Figure 6. Maximum Peak Output Voltage vs Supply Voltage

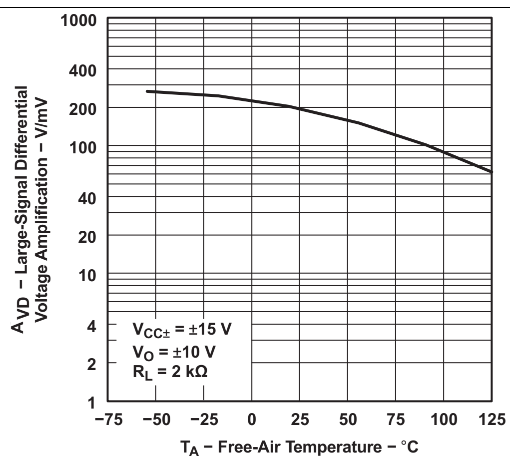


Figure 7. Large-Signal Differential Voltage Amplification vs Free-Air Temperature

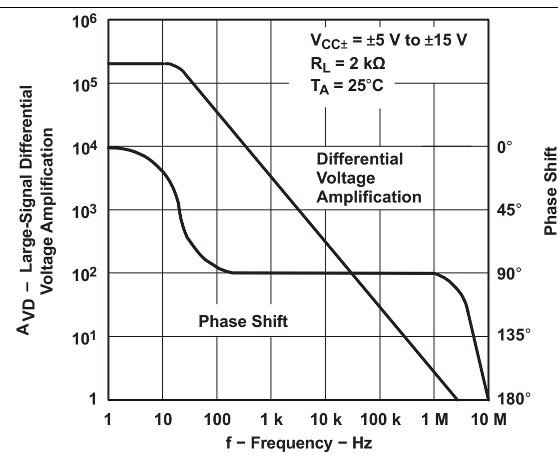


Figure 8. Large-Signal Differential Voltage Amplification and Phase Shift vs Frequency

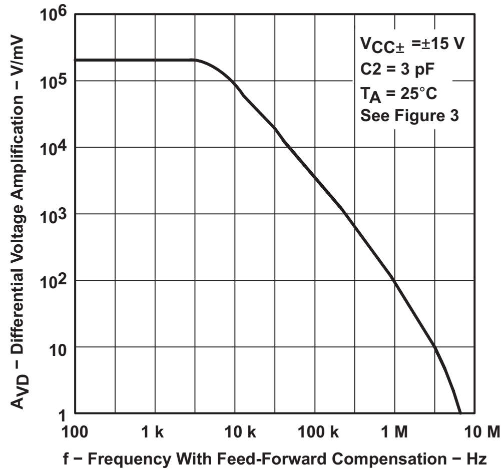


Figure 9. Differential Voltage Amplification vs Frequency with Feed-Forward Compensation

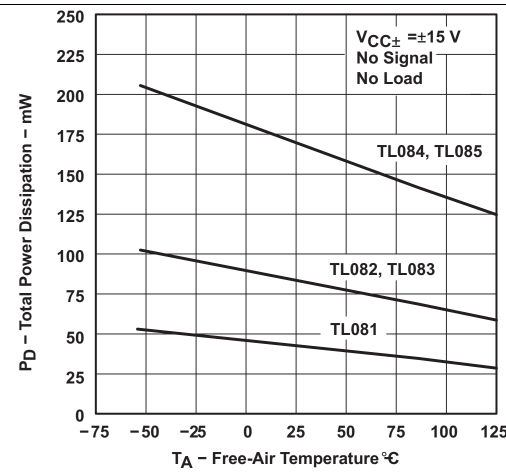


Figure 10. Total Power Dissipation vs Free-Air Temperature

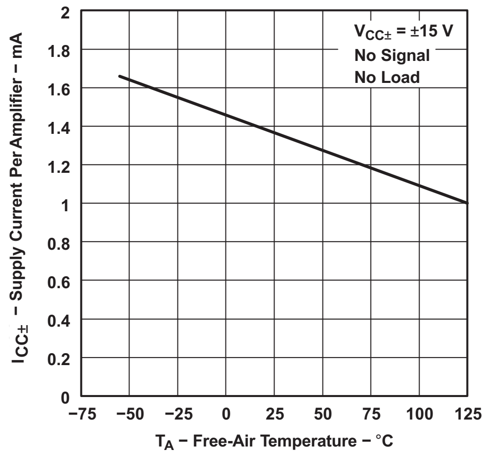


Figure 11. Supply Current per Amplifier vs Free-Air Temperature

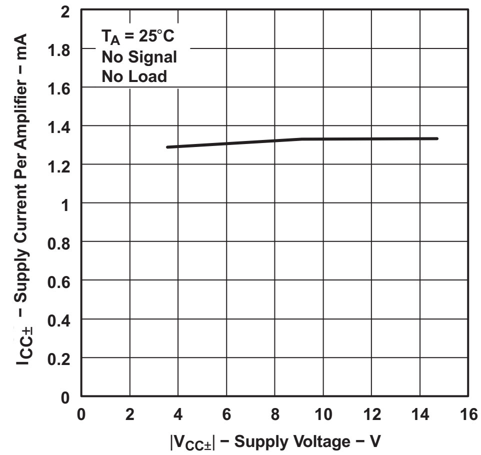


Figure 12. Supply Current per Amplifier vs Supply Voltage

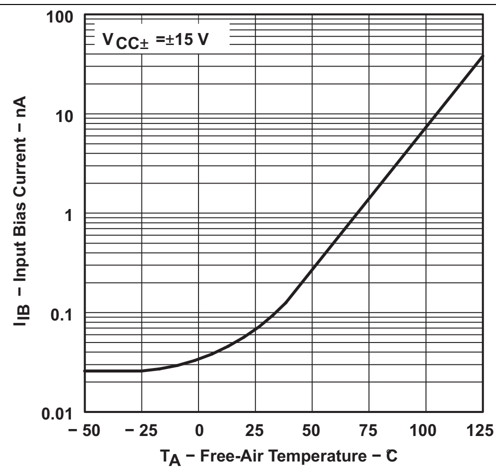


Figure 13. Input Bias Current vs Free-Air Temperature

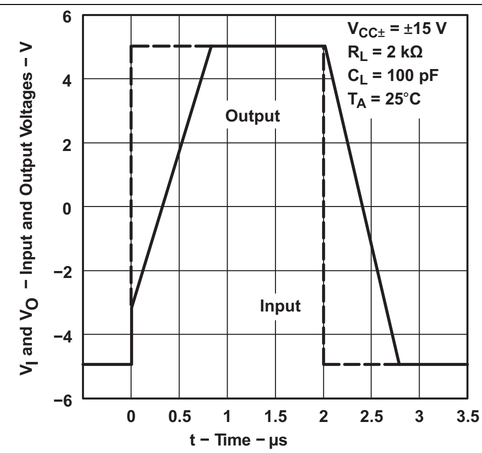


Figure 14. Voltage-Follower Large-Signal Pulse Response

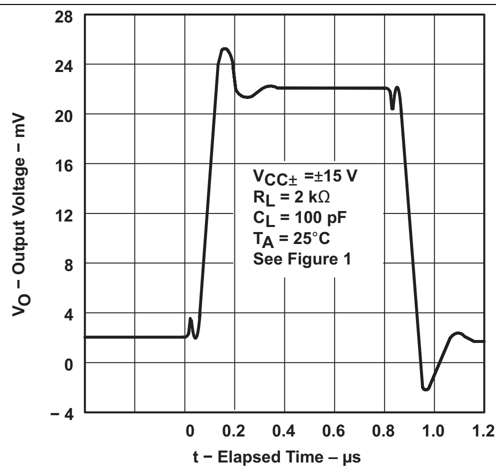


Figure 15. Output Voltage vs Elapsed Time

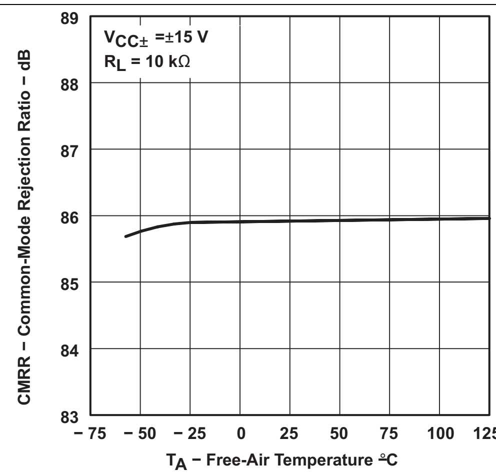


Figure 16. Common-Mode Rejection Ratio vs Free-Air Temperature

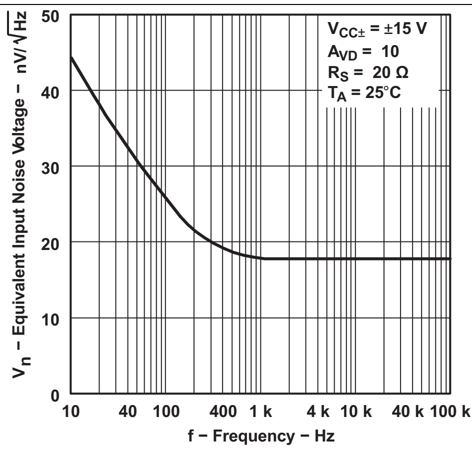


Figure 17. Equivalent Input Noise Voltage vs Frequency

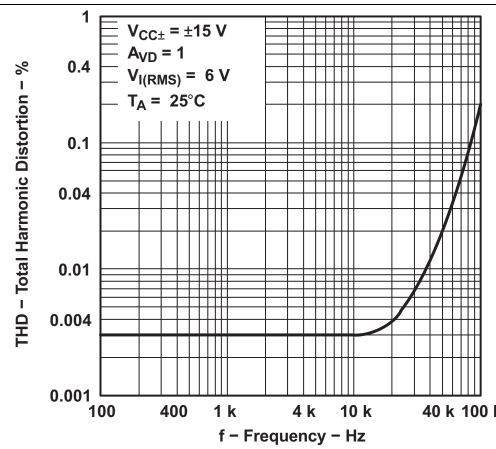


Figure 18. Total Harmonic Distortion vs Frequency

Thermally Enhanced, Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 100 μΩ Current Conductor

Not for New Design

These parts are in production but have been determined to be NOT FOR NEW DESIGN. This classification indicates that sale of this device is currently restricted to existing customer applications. The device should not be purchased for new design applications because obsolescence in the near future is probable. Samples are no longer available.

Date of status change: June 5, 2017

Recommended Substitutions:

For existing customer transition, and for new customers or new applications, use ACS770xCB.

NOTE: For detailed information on purchasing options, contact your local Allegro field applications engineer or sales representative.

ACS758xCB

Thermally Enhanced, Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 100 μΩ Current Conductor

FEATURES AND BENEFITS

- Industry-leading noise performance through proprietary amplifier and filter design techniques
- Integrated shield greatly reduces capacitive coupling from current conductor to die due to high dV/dt signals, and prevents offset drift in high-side, high-voltage applications
- Total output error improvement through gain and offset trim over temperature
- Small package size, with easy mounting capability
- Monolithic Hall IC for high reliability
- Ultralow power loss: 100 μΩ internal conductor resistance
- Galvanic isolation allows use in economical, high-side current sensing in high-voltage systems
- AEC-Q100 qualified

Continued on the next page…

TÜV America
Certificate Number:
U8V 14 05 54214 028
UL Certified
File No.: E316429

PACKAGE: 5-Pin CB Package

PFF Leadform **PSF Leadform** **PSS Leadform**

DESCRIPTION

The Allegro™ ACS758 family of current sensor ICs provides economical and precise solutions for AC or DC current sensing. Typical applications include motor control, load detection and management, power supply and DC-to-DC converter control, inverter control, and overcurrent fault detection.

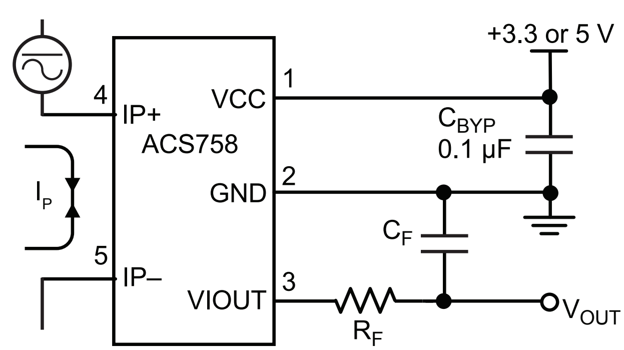
The device consists of a precision, low-offset linear Hall circuit with a copper conduction path located near the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional output voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy at the factory.

High-level immunity to current conductor dV/dt and stray electric fields, offered by Allegro proprietary integrated shield technology, provides low output voltage ripple and low offset drift in high-side, high-voltage applications.

The output of the device has a positive slope ($>V_{CC}/2$) when an increasing current flows through the primary copper conduction path (from terminal 4 to terminal 5), which is the path used for current sampling. The internal resistance of this conductive path is 100 μΩ typical, providing low power loss.

The thickness of the copper conductor allows survival of the device at high overcurrent conditions. The terminals of the conductive path are electrically isolated from the signal leads

Continued on the next page…



Application 1: The ACS758 outputs an analog signal, V_{OUT}, that varies linearly with the uni- or bi-directional AC or DC primary sampled current, I_P, within the range specified. C_F is for optimal noise management, with values that depend on the application.

Typical Application

FEATURES AND BENEFITS (CONTINUED)

- 3.0 to 5.5 V, single supply operation
- 120 kHz typical bandwidth
- 3 μs output rise time in response to step input current
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis

DESCRIPTION (CONTINUED)

(pins 1 through 3). This allows the ACS758 family of sensor ICs to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The device is fully calibrated prior to shipment from the factory. The ACS758 family is lead (Pb) free. All leads are plated with 100% matte tin, and there is no Pb inside the package. The heavy gauge leadframe is made of oxygen-free copper.

Selection Guide

Part Number [1]	Package		Primary Sampled Current, I_P (A)	Sensitivity Sens (Typ.) (mV/A)	Current Directionality	T_{OP} (°C)	Packing [2]
	Terminals	Signal Pins					
ACS758LCB-050B-PFF-T	Formed	Formed	±50	40	Bidirectional	–40 to 150	34 pieces per tube
ACS758LCB-050U-PFF-T	Formed	Formed	50	60	Unidirectional		
ACS758LCB-100B-PFF-T	Formed	Formed	±100	20	Bidirectional		
ACS758LCB-100B-PSF-T	Straight	Formed	±100	20	Bidirectional		
ACS758LCB-100U-PFF-T	Formed	Formed	100	40	Unidirectional		
ACS758KCB-150B-PFF-T	Formed	Formed	±150	13.3	Bidirectional	–40 to 125	
ACS758KCB-150U-PSF-T	Straight	Formed	150	26.7	Unidirectional		
ACS758KCB-150B-PSS-T	Straight	Straight	±150	13.3	Bidirectional		
ACS758KCB-150U-PFF-T	Formed	Formed	150	26.7	Unidirectional		
ACS758ECB-200B-PFF-T	Formed	Formed	±200	10	Bidirectional	–40 to 85	
ACS758ECB-200B-PSF-T	Straight	Formed	±200	10	Bidirectional		
ACS758ECB-200U-PSF-T	Straight	Formed	200	20	Unidirectional		
ACS758ECB-200B-PSS-T	Straight	Straight	±200	10	Bidirectional		
ACS758ECB-200U-PFF-T	Formed	Formed	200	20	Unidirectional		

[1] Additional leadform options available for qualified volumes.
[2] Contact Allegro for additional packing options.

Allegro MicroSystems, LLC
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Characteristic	Symbol	Notes	Rating	Units
Forward Supply Voltage	V_{CC}		8	V
Reverse Supply Voltage	V_{RCC}		–0.5	V
Forward Output Voltage	V_{IOUT}		28	V
Reverse Output Voltage	V_{RIOUT}		–0.5	V
Output Source Current	$I_{OUT(Source)}$	VIOUT to GND	3	mA
Output Sink Current	$I_{OUT(Sink)}$	VCC to VIOUT	1	mA
Nominal Operating Ambient Temperature	T_{OP}	Range E	–40 to 85	°C
		Range K	–40 to 125	°C
		Range L	–40 to 150	°C
Maximum Junction	T_J(max)		165	°C
Storage Temperature	T_{stg}		–65 to 165	°C

ISOLATION CHARACTERISTICS

Characteristic	Symbol	Notes	Rating	Unit
Dielectric Strength Test Voltage [1]	V_{ISO}	Agency type-tested for 60 seconds per UL standard 60950-1, 2nd Edition	4800	VAC
Working Voltage for Basic Isolation	V_{WFSI}	For basic (single) isolation per UL standard 60950-1, 2nd Edition	990	VDC or V_{pk}
			700	V_{rms}
Working Voltage for Reinforced Isolation	V_{WFRI}	For reinforced (double) isolation per UL standard 60950-1, 2nd Edition	636	VDC or V_{pk}
			450	V_{rms}

[1] Allegro does not conduct 60-second testing. It is done only during the UL certification process.

THERMAL CHARACTERISTICS: May require derating at maximum conditions

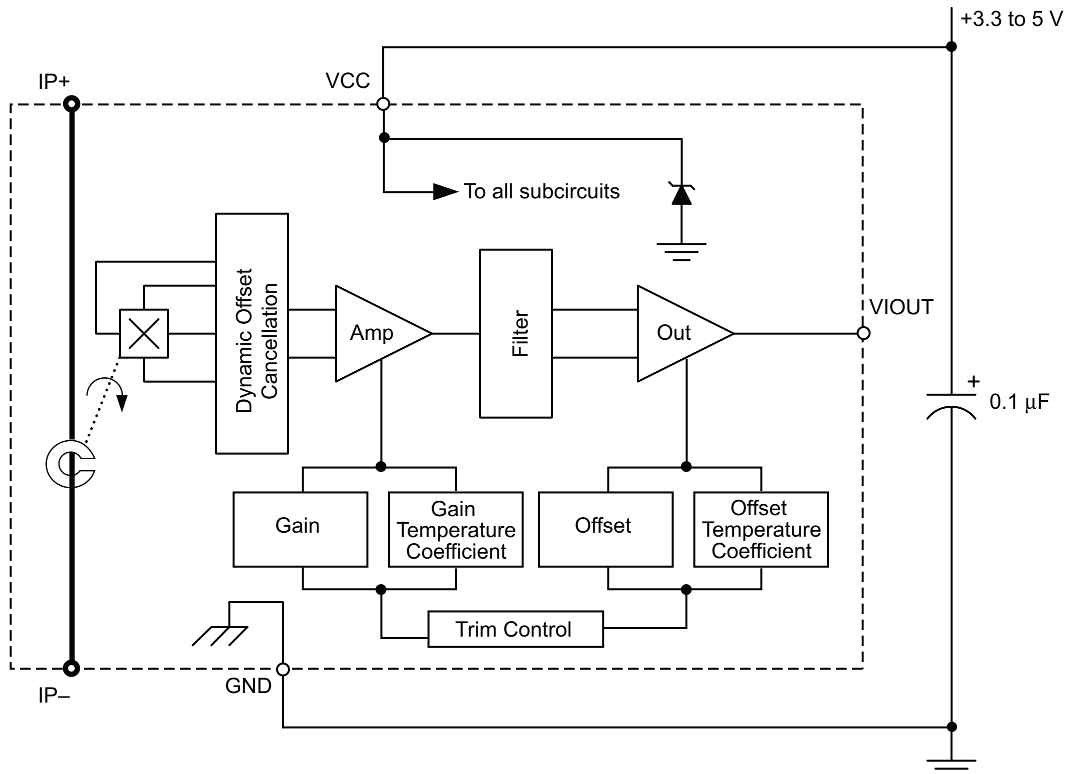
Characteristic	Symbol	Test Conditions [1]	Value	Unit
Package Thermal Resistance	$R_{θJA}$	Mounted on the Allegro evaluation board with 2800 mm^2 (1400 mm^2 on component side and 1400 mm^2 on opposite side) of 4 oz. copper connected to the primary leadframe and with thermal vias connecting the copper layers. Performance is based on current flowing through the primary leadframe and includes the power consumed by the PCB.	7	°C/W

[1] Additional thermal information available on the Allegro website

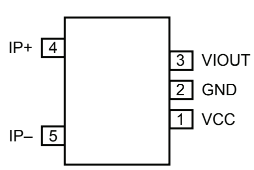
TYPICAL OVERCURRENT CAPABILITIES [2][3]

Characteristic	Symbol	Notes	Rating	Units
Overcurrent	I_{POC}	T_A = 25°C, 1 second duration, 1% duty cycle	1200	A
		T_A = 85°C, 1 second duration, 1% duty cycle	900	A
		T_A = 150°C, 1 second duration, 1% duty cycle	600	A

[2] Test was done with Allegro evaluation board. The maximum allowed current is limited by T_J(max) only.
[3] For more overcurrent profiles, please see FAQ on the Allegro website, www.allegromicro.com.



Functional Block Diagram



Pinout Diagram

Terminal List Table

Number	Name	Description
1	VCC	Device power supply terminal
2	GND	Signal ground terminal
3	VIOUT	Analog output signal
4	IP+	Terminal for current being sampled
5	IP–	Terminal for current being sampled

COMMON OPERATING CHARACTERISTICS [1]: Valid at T_{OP} = –40°C to 150°C and V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Supply Voltage [2]	V_{CC}		3	5.0	5.5	V
Supply Current	I_{CC}	Output open	–	10	13.5	mA
Power-On Delay	t_{POD}	T_A = 25°C	–	10	–	µs
Rise Time [3]	t_r	I_P step = 60% of I_P+, 10% to 90% rise time, T_A = 25°C, C_{OUT} = 0.47 nF	–	3	–	µs
Propagation Delay Time [3]	t_{PROP}	T_A = 25°C, C_{OUT} = 0.47 nF	–	1	–	µs
Response Time	$t_{RESPONSE}$	Measured as sum of t_{PROP} and t_r	–	4	–	µs
Internal Bandwidth [4]	BW_i	–3 dB; T_A = 25°C, C_{OUT} = 0.47 nF	–	120	–	kHz
Output Load Resistance	$R_{LOAD(MIN)}$	VIOUT to GND	4.7	–	–	kΩ
Output Load Capacitance	$C_{LOAD(MAX)}$	VIOUT to GND	–	–	10	nF
Primary Conductor Resistance	$R_{PRIMARY}$	T_A = 25°C	–	100	–	µΩ
Symmetry [3]	E_{SYM}	Over half-scale of I_P	99	100	101	%
Quiescent Output Voltage [5]	$V_{IOUT(QBI)}$	Bidirectional variant, I_P = 0 A, T_A = 25°C	–	$V_{CC}/2$	–	V
	$V_{IOUT(QUNI)}$	Unidirectional variant, I_P = 0 A, T_A = 25°C, $V_{IOUT(QUNI)}$ is ratiometric to V_{CC}	–	0.6	–	V
Ratiometry [3]	V_{RAT}	V_{CC} = 4.5 to 5.5 V	–	100	–	%

1 Device is factory-trimmed at 5 V, for optimal accuracy.
2 Devices are programmed for maximum accuracy at 5.0 V V_{CC} levels. The device contains ratiometry circuits that accurately alter the 0 A Output Voltage and Sensitivity level of the device in proportion to the applied V_{CC} level. However, as a result of minor nonlinearities in the ratiometry circuit additional output error will result when V_{CC} varies from the 5 V V_{CC} level. Customers that plan to operate the device from a 3.3 V regulated supply should contact their local Allegro sales representative regarding expected device accuracy levels under these bias conditions.
3 See Characteristic Definitions section of this datasheet.
4 Calculated using the formula BW_i = 0.35 / t_r.
5 $V_{IOUT(Q)}$ may drift over the lifetime of the device by as much as ±25 mV.

*X*050B PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 150°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		–50	–	50	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	40	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	39.4	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	41	–	mV/A
Noise [2]	V_{NOISE}	T_A = 25°C, 10 nF on VIOUT pin to GND	–	10	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	–1	–	1	%
Electrical Offset Voltage [3]	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 150°C	–	±15	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±35	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 50 A	–	100	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	–1.2	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	2	–	%

[1] See Characteristic Performance Data page for parameter distributions over temperature range.
[2] ±3 sigma noise voltage.
[3] $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 2.5 V.
[4] Percentage of I_P. Output filtered.

*X*050U PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 150°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		0	–	50	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	60	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	59	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	61	–	mV/A
Noise [2]	V_{NOISE}	T_A = 25°C, 10 nF on VIOUT pin to GND	–	15	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	–1	–	1	%
Electrical Offset Voltage [3]	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 150°C	–	±20	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±40	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 50 A	–	100	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	–1.2	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	2	–	%

[1] See Characteristic Performance Data page for parameter distributions over temperature range.
[2] ±3 sigma noise voltage.
[3] $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 0.6 V.
[4] Percentage of I_P. Output filtered.

*X*100B PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 150°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		–100	–	100	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	20	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	19.75	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	20.5	–	mV/A
Noise [2]	V_{NOISE}	T_A= 25°C, 10 nF on VIOUT pin to GND	–	6	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	–1.25	–	1.25	%
Electrical Offset Voltage [3]	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 150°C	–	±20	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±20	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 100 A	–	150	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	–1.3	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	2.4	–	%

1 See Characteristic Performance Data page for parameter distributions over temperature range.
2 ±3 sigma noise voltage.
3 $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 2.5 V.
4 Percentage of I_P. Output filtered.

*X*100U PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 150°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		0	–	100	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	40	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	39.5	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	41	–	mV/A
Noise [2]	V_{NOISE}	T_A= 25°C, 10 nF on VIOUT pin to GND	–	12	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	–1.25	–	1.25	%
Electrical Offset Voltage [3]	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 150°C	–	±20	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±20	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 100 A	–	150	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 150°C	–	–1.3	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	2.4	–	%

1 See Characteristic Performance Data page for parameter distributions over temperature range.
2 ±3 sigma noise voltage.
3 $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 0.6 V.
4 Percentage of I_P. Output filtered.

*X*150B PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 125°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		–150	–	150	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	13.3	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 125°C	–	13.1	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	13.5	–	mV/A
Noise [2]	V_{NOISE}	T_A= 25°C, 10 nF on VIOUT pin to GND	–	4	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	– 1	–	1	%
Electrical Offset Voltage [3]	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 125°C	–	±14	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±24	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 150 A	–	205	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 125°C	–	–1.8	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	1.6	–	%

[1] See Characteristic Performance Data page for parameter distributions over temperature range.
[2] ±3 sigma noise voltage.
[3] $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 2.5 V.
[4] Percentage of I_P. Output filtered.

*X*150U PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 125°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		0	–	150	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	26.6	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 125°C	–	26.6	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	27.4	–	mV/A
Noise [2]	V_{NOISE}	T_A= 25°C, 10 nF on VIOUT pin to GND	–	8	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	– 1	–	1	%
Electrical Offset Voltage [3]	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 125°C	–	±14	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±24	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 150 A	–	205	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 125°C	–	–1.8	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	1.6	–	%

[1] See Characteristic Performance Data page for parameter distributions over temperature range.
[2] ±3 sigma noise voltage.
[3] $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 0.6 V.
[4] Percentage of I_P. Output filtered.

*X*200B PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 85°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		–200	–	200	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	10	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 85°C	–	9.88	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	10.13	–	mV/A
Noise [2]	V_{NOISE}	T_A= 25°C, 10 nF on VIOUT pin to GND	–	3	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	– 1	–	1	%
Electrical Offset Voltage [3][	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 85°C	–	±15	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±25	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 200 A	–	230	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 85°C	–	–1.2	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	1.2	–	%

1 See Characteristic Performance Data page for parameter distributions over temperature range.
2 ±3 sigma noise voltage.
3 $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 2.5 V.
4 Percentage of I_P. Output filtered.

*X*200U PERFORMANCE CHARACTERISTICS [1]: T_{OP} = –40°C to 85°C, V_{CC} = 5 V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Primary Sampled Current	I_P		0	–	200	A
Sensitivity	$Sens_{TA}$	Full scale of I_P applied for 5 ms, T_A = 25°C	–	20	–	mV/A
	$Sens_{(TOP)HT}$	Full scale of I_P applied for 5 ms, T_{OP} = 25°C to 85°C	–	19.7	–	mV/A
	$Sens_{(TOP)LT}$	Full scale of I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	20.3	–	mV/A
Noise [2]	V_{NOISE}	T_A= 25°C, 10 nF on VIOUT pin to GND	–	6	–	mV
Nonlinearity	E_{LIN}	Up to full scale of I_P, I_P applied for 5 ms	– 1	–	1	%
Electrical Offset Voltage [3]	$V_{OE(TA)}$	I_P = 0 A, T_A = 25°C	–	±5	–	mV
	$V_{OE(TOP)HT}$	I_P = 0 A, T_{OP} = 25°C to 85°C	–	±20	–	mV
	$V_{OE(TOP)LT}$	I_P = 0 A, T_{OP} = –40°C to 25°C	–	±35	–	mV
Magnetic Offset Error	I_{ERROM}	I_P = 0 A, T_A = 25°C, after excursion of 200 A	–	230	–	mA
Total Output Error [4]	$E_{TOT(HT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = 25°C to 85°C	–	–1.2	–	%
	$E_{TOT(LT)}$	Over full scale of I_P, I_P applied for 5 ms, T_{OP} = –40°C to 25°C	–	1.2	–	%

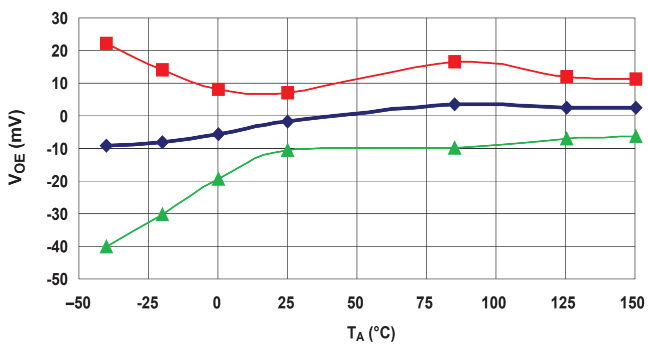
1 See Characteristic Performance Data page for parameter distributions over temperature range.
2 ±3 sigma noise voltage.
3 $V_{OE(TOP)}$ drift is referred to ideal $V_{IOUT(Q)}$ = 0.6 V.
4 Percentage of I_P. Output filtered.

CHARACTERISTIC PERFORMANCE DATA

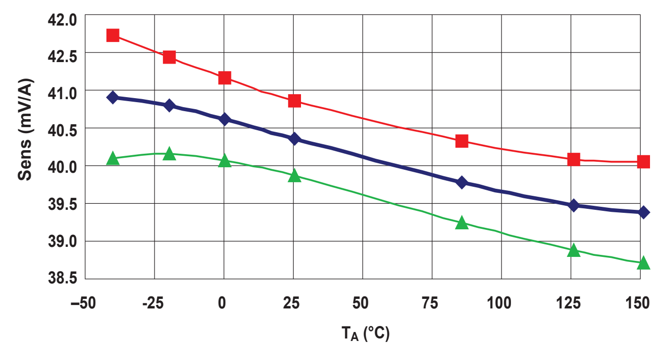
Data taken using the ACS758LCB-050B

Accuracy Data

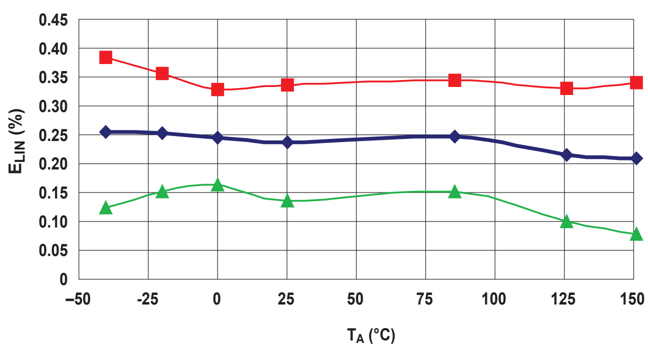
Electrical Offset Voltage versus Ambient Temperature



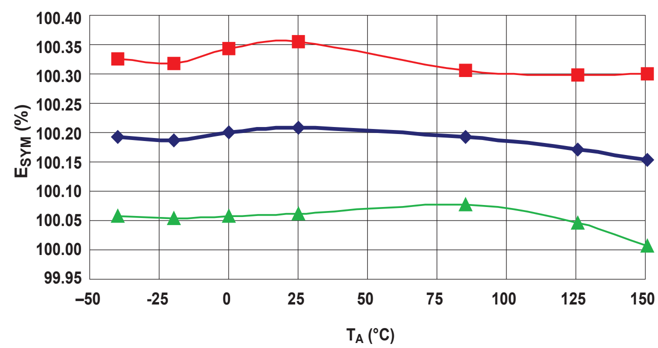
Sensitivity versus Ambient Temperature



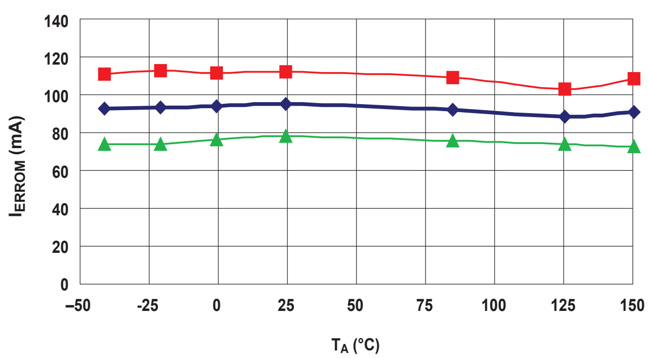
Nonlinearity versus Ambient Temperature



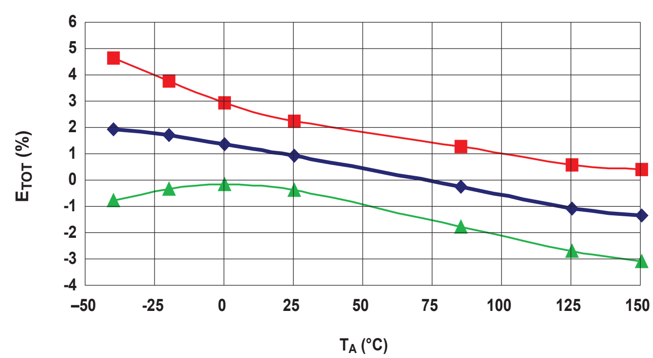
Symmetry versus Ambient Temperature



Magnetic Offset Error versus Ambient Temperature



Total Output Error versus Ambient Temperature



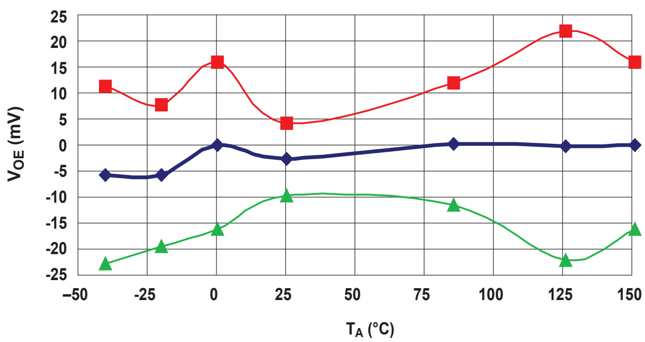
Typical Maximum Limit — Mean — Typical Minimum Limit

CHARACTERISTIC PERFORMANCE DATA

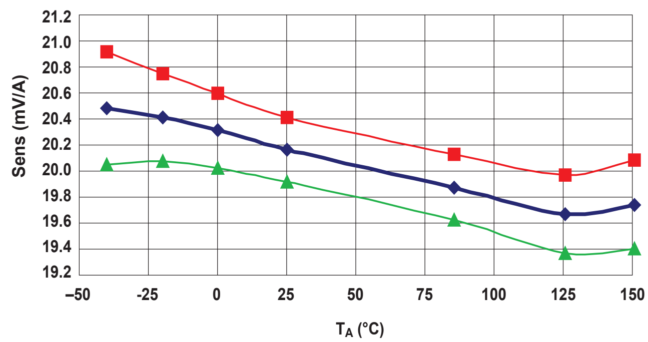
Data taken using the ACS758LCB-100B

Accuracy Data

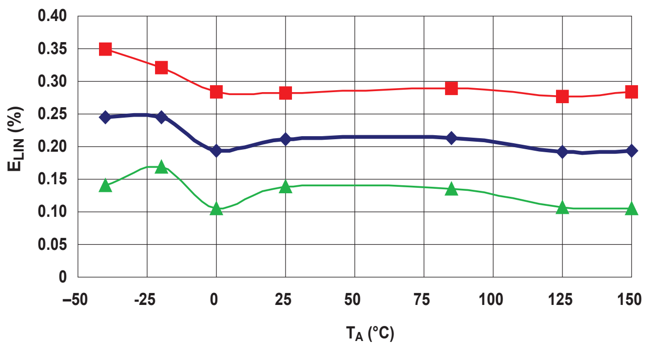
Electrical Offset Voltage versus Ambient Temperature



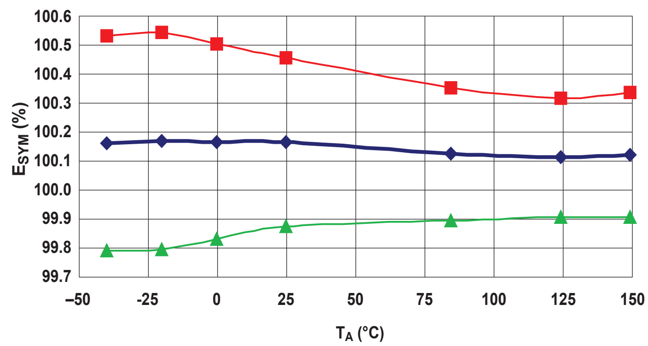
Sensitivity versus Ambient Temperature



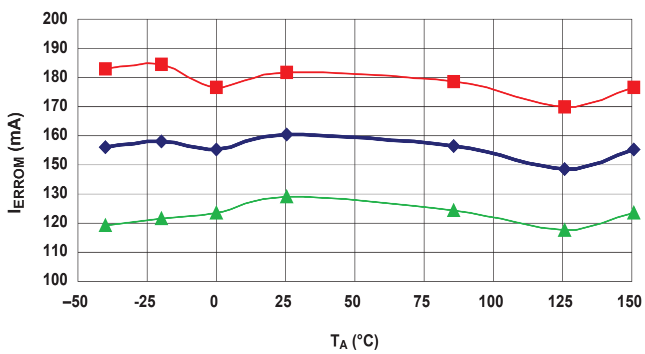
Nonlinearity versus Ambient Temperature



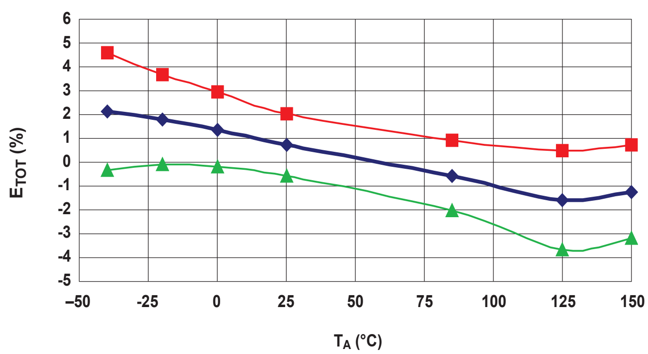
Symmetry versus Ambient Temperature



Magnetic Offset Error versus Ambient Temperature



Total Output Error versus Ambient Temperature

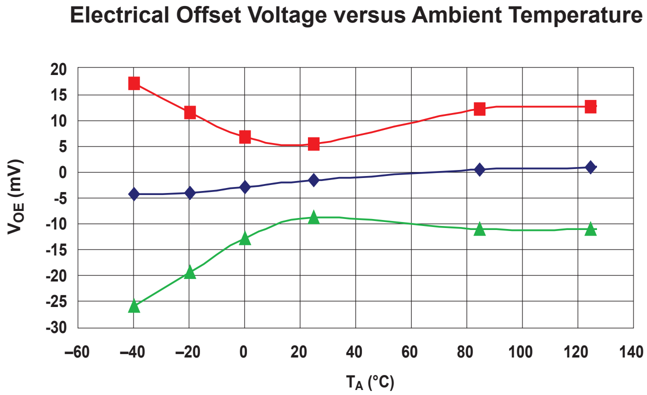


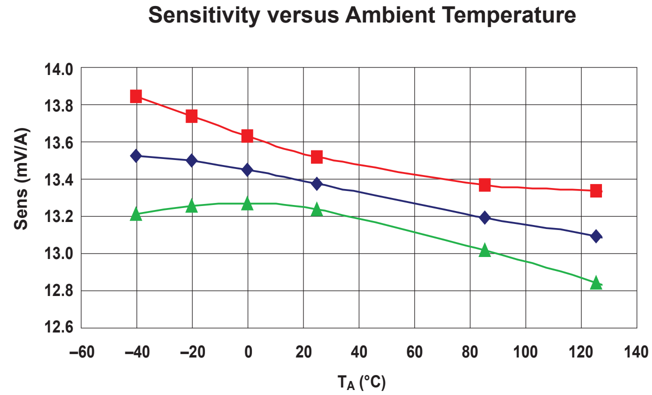
Typical Maximum Limit — Mean — Typical Minimum Limit

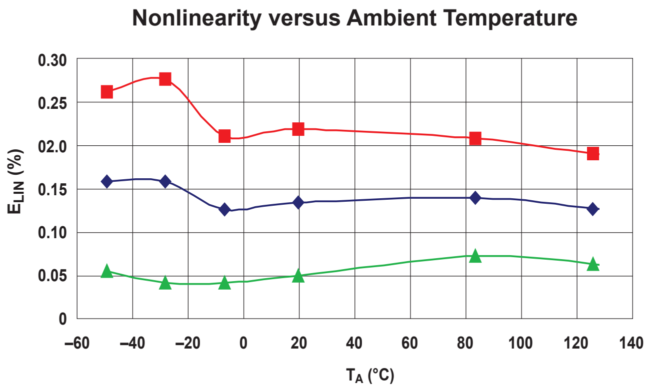
CHARACTERISTIC PERFORMANCE DATA

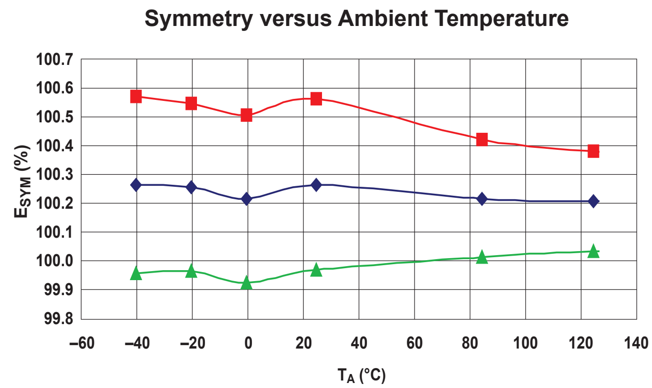
Data taken using the ACS758KCB-150B

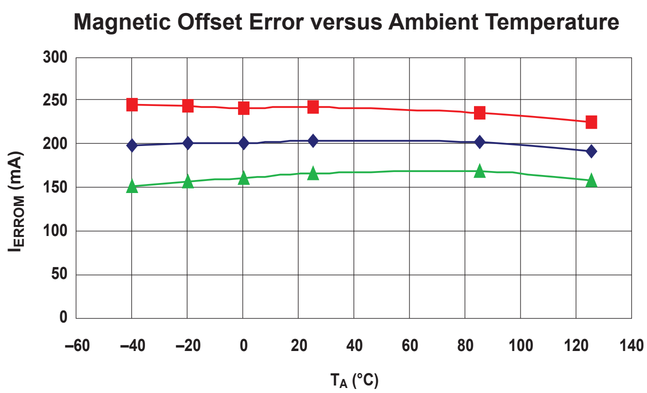
Accuracy Data

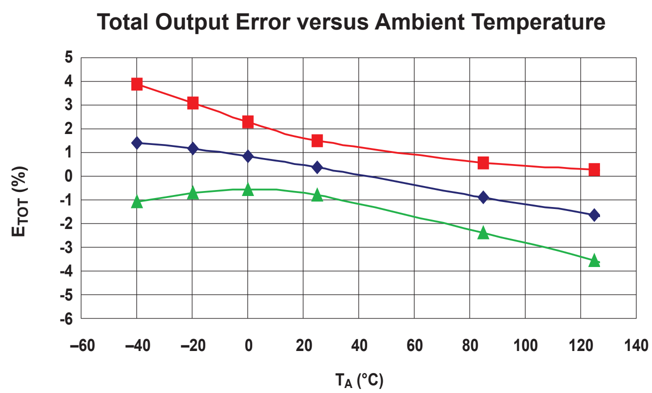












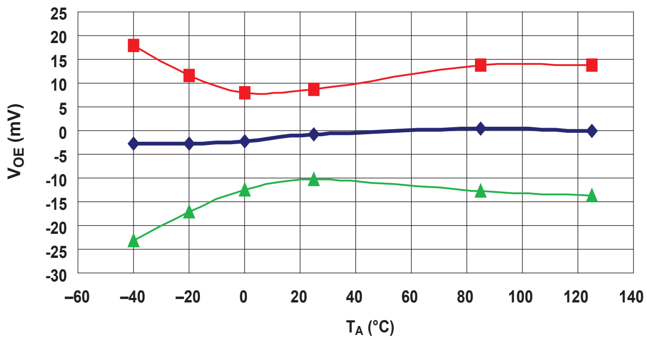
Typical Maximum Limit — Mean — Typical Minimum Limit

CHARACTERISTIC PERFORMANCE DATA

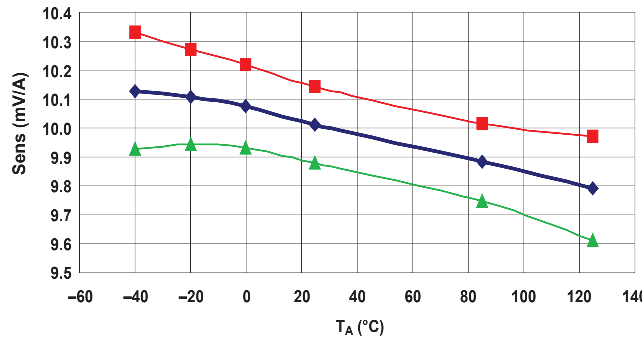
Data taken using the ACS758ECB-200B

Accuracy Data

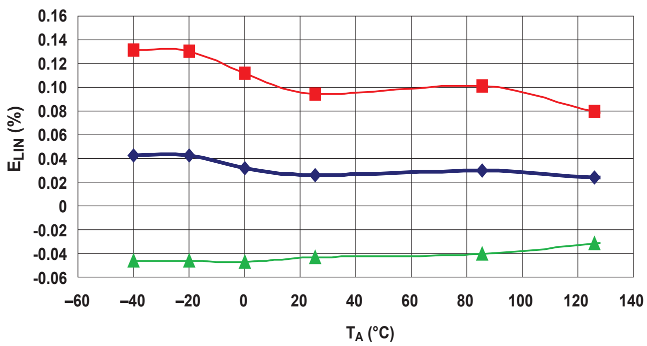
Electrical Offset Voltage versus Ambient Temperature



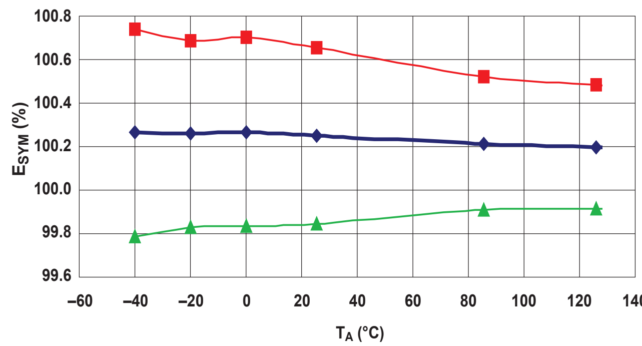
Sensitivity versus Ambient Temperature



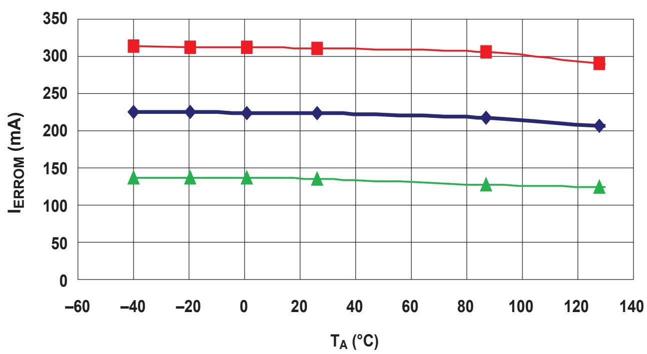
Nonlinearity versus Ambient Temperature



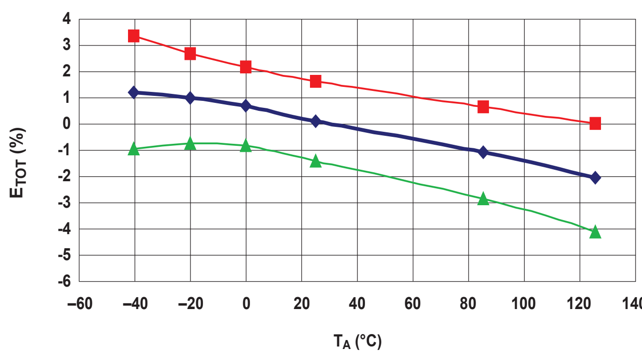
Symmetry versus Ambient Temperature



Magnetic Offset Error versus Ambient Temperature



Total Output Error versus Ambient Temperature



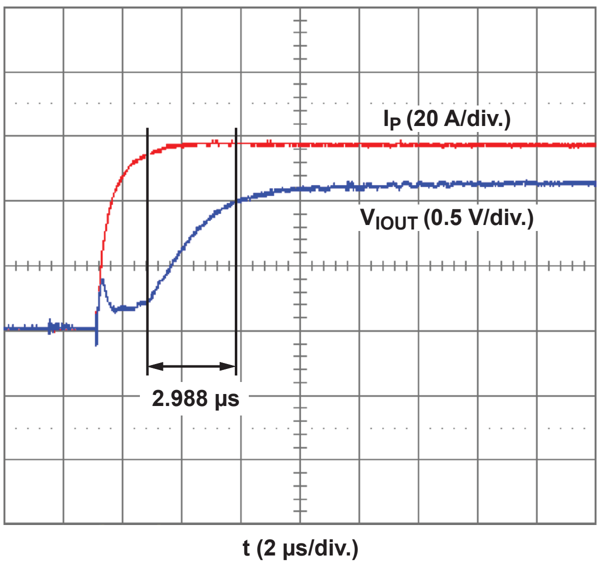
Typical Maximum Limit — Mean — Typical Minimum Limit

CHARACTERISTIC PERFORMANCE DATA

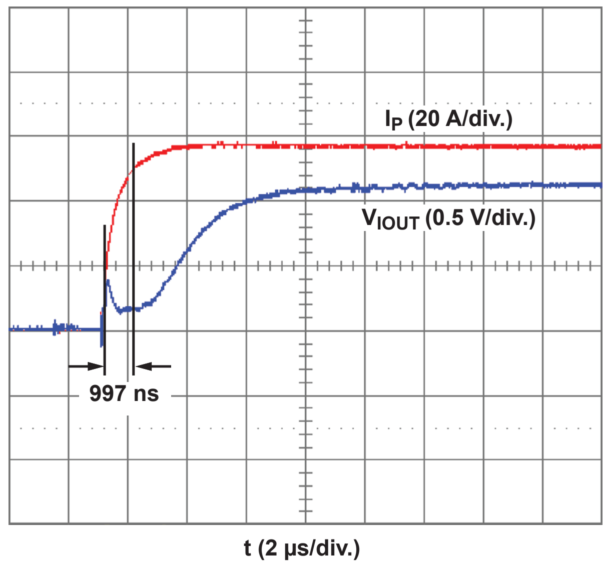
Data taken using the ACS758LCB-100B

Timing Data

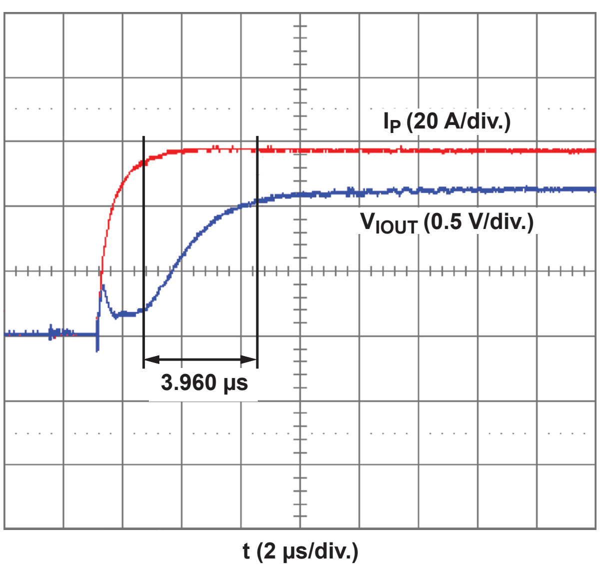
Rise Time



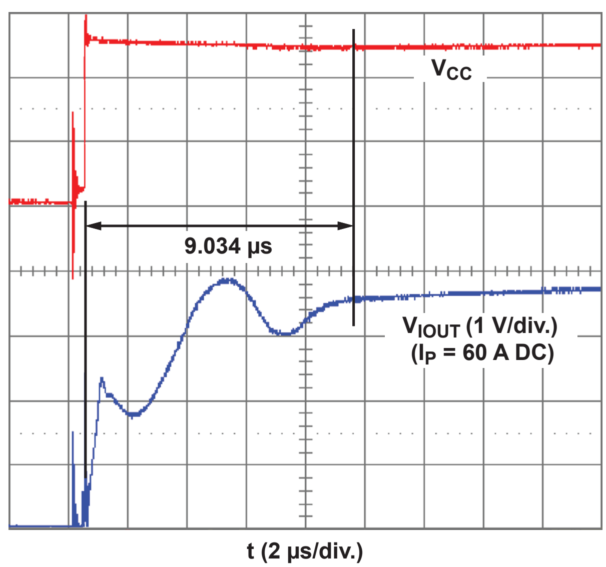
Propagation Delay Time



Response Time



Power-on Delay



CHARACTERISTIC DEFINITIONS

Definitions of Accuracy Characteristics

Sensitivity (Sens). The change in device output in response to a 1 A change through the primary conductor. The sensitivity is the product of the magnetic circuit sensitivity (G/A) and the linear IC amplifier gain (mV/G). The linear IC amplifier gain is programmed at the factory to optimize the sensitivity (mV/A) for the half-scale current of the device.

Noise (V_{NOISE}). The noise floor is derived from the thermal and shot noise observed in Hall elements. Dividing the noise (mV) by the sensitivity (mV/A) provides the smallest current that the device is able to resolve.

Nonlinearity (E_{LIN}). The degree to which the voltage output from the IC varies in direct proportion to the primary current through its half-scale amplitude. Nonlinearity in the output can be attributed to the saturation of the flux concentrator approaching the half-scale current. The following equation is used to derive the linearity:

$$100\left\{1-\left[\frac{\Delta\text{ gain}\times\%\text{ sat }(V_{IOUT\_half\text{-}scale\ amperes}-V_{IOUT(Q)})}{2\,(V_{IOUT\_quarter\text{-}scale\ amperes}-V_{IOUT(Q)})}\right]\right\}$$

where

Δ gain = the gain variation as a function of temperature changes from 25°C,

% sat = the percentage of saturation of the flux concentrator, which becomes significant as the current being sampled approaches half-scale $\pm I_P$, and

$V_{IOUT\_half\text{-}scale\ amperes}$ = the output voltage (V) when the sampled current approximates half-scale $\pm I_P$.

Symmetry (E_{SYM}). The degree to which the absolute voltage output from the IC varies in proportion to either a positive or negative half-scale primary current. The following equation is used to derive symmetry:

$$100\left(\frac{V_{IOUT\_+half\text{-}scale\ amperes}-V_{IOUT(Q)}}{V_{IOUT(Q)}-V_{IOUT\_-half\text{-}scale\ amperes}}\right)$$

Ratiometry. The device features a ratiometric output. This means that the quiescent voltage output, V_{IOUTQ}, and the magnetic sensitivity, Sens, are proportional to the supply voltage, V_{CC}.

The ratiometric change (%) in the quiescent voltage output is defined as:

$$\Delta V_{IOUTQ(\Delta V)}=\frac{V_{IOUTQ(V_{CC})}\,/\,V_{IOUTQ(5V)}}{V_{CC}\,/\,5\text{ V}}\times 100\%$$

and the ratiometric change (%) in sensitivity is defined as:

$$\Delta\text{Sens}_{(\Delta V)}=\frac{\text{Sens}_{(V_{CC})}\,/\,\text{Sens}_{(5V)}}{V_{CC}\,/\,5\text{ V}}\times 100\%$$

Quiescent output voltage ($V_{IOUT(Q)}$). Quiescent output voltage ($V_{IOUT(Q)}$). The output of the device when the primary current is zero. For bidirectional devices, it nominally remains at $V_{CC}/2$. Thus, V_{CC} = 5 V translates into $V_{IOUT(QBI)}$ = 2.5 V. For unidirectional devices, it nominally remains at 0.12 × V_{CC}. Thus, V_{CC} = 5 V translates into $V_{IOUT(QUNI)}$ = 0.6 V. Variation in $V_{IOUT(Q)}$ can be attributed to the resolution of the Allegro linear IC quiescent voltage trim, magnetic hysteresis, and thermal drift.

Electrical offset voltage (V_{OE}). The deviation of the device output from its ideal quiescent value of $V_{CC}/2$ for bidirectional and 0.1 × V_{CC} for unidirectional devices, due to nonmagnetic causes.

Magnetic offset error (I_{ERROM}). The magnetic offset is due to the residual magnetism (remnant field) of the core material. The magnetic offset error is highest when the magnetic circuit has been saturated, usually when the device has been subjected to a full-scale or high-current overload condition. The magnetic offset is largely dependent on the material used as a flux concentrator. The larger magnetic offsets are observed at the lower operating temperatures.

Total Output Error (E_{TOT}). The maximum deviation of the actual output from its ideal value, also referred to as *accuracy*, illustrated graphically in the output voltage versus current chart on the following page.

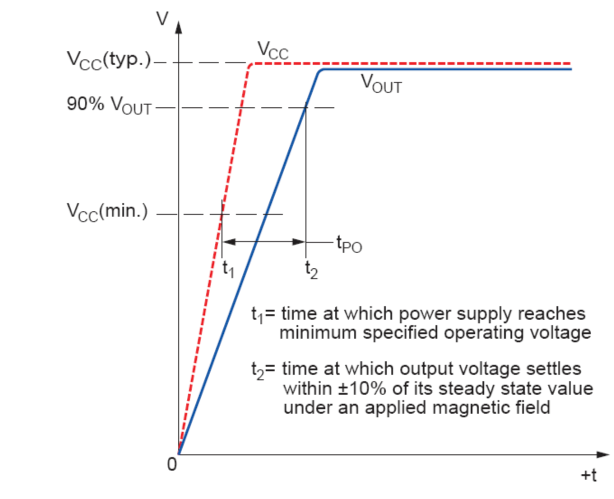
E_{TOT} is divided into four areas:

- **0 A at 25°C.** Accuracy at the zero current flow at 25°C, without the effects of temperature.
- **0 A over Δ temperature.** Accuracy at the zero current flow including temperature effects.
- **Half-scale current at 25°C.** Accuracy at the the half-scale current at 25°C, without the effects of temperature.
- **Half-scale current over Δ temperature.** Accuracy at the half-scale current flow including temperature effects.

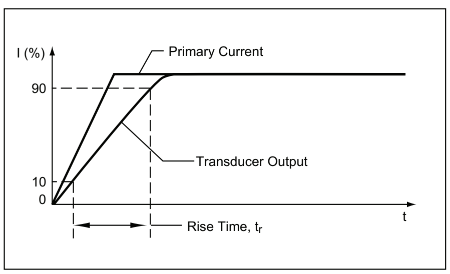
Definitions of Dynamic Response Characteristics

Power-On Time (t_{PO}). When the supply is ramped to its operating voltage, the device requires a finite time to power its internal components before responding to an input magnetic field.

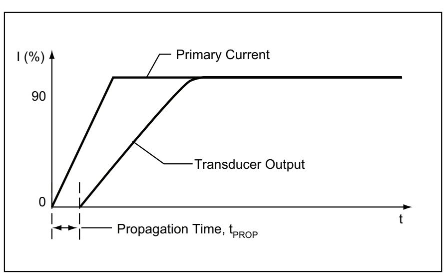
Power-On Time, t_{PO} , is defined as the time it takes for the output voltage to settle within ±10% of its steady state value under an applied magnetic field, after the power supply has reached its minimum specified operating voltage, V_{CC}(min), as shown in the chart at right.



Rise time (t_r). The time interval between a) when the device reaches 10% of its full scale value, and b) when it reaches 90% of its full scale value. The rise time to a step response is used to derive the bandwidth of the device, in which $f(–3\text{ dB}) = 0.35 / t_r$. Both t_r and $t_{RESPONSE}$ are detrimentally affected by eddy current losses observed in the conductive IC ground plane.

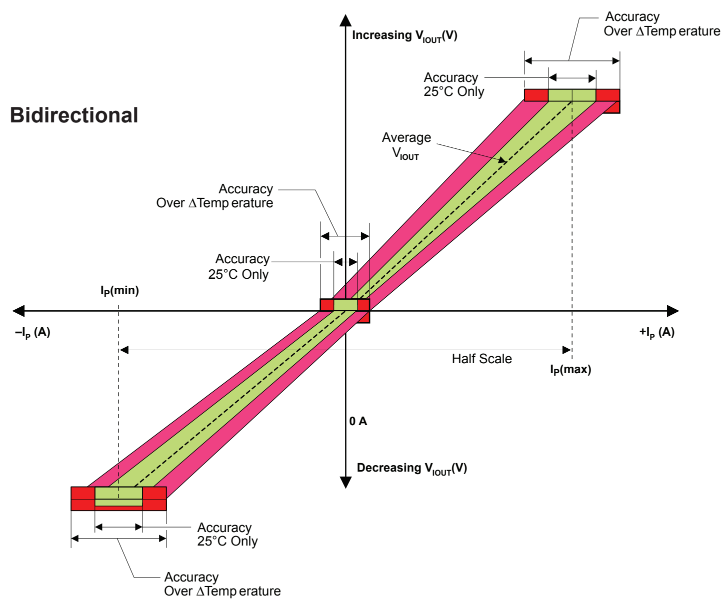


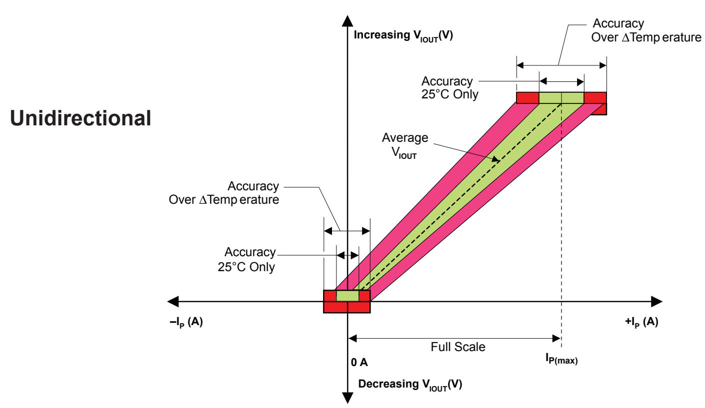
Propagation delay (t_{PROP}). The time required for the device output to reflect a change in the primary current signal. Propagation delay is attributed to inductive loading within the linear IC package, as well as in the inductive loop formed by the primary conductor geometry. Propagation delay can be considered as a fixed time offset and may be compensated.



Output Voltage versus Sampled Current

Total Output Error at 0 A and at Half-Scale Current





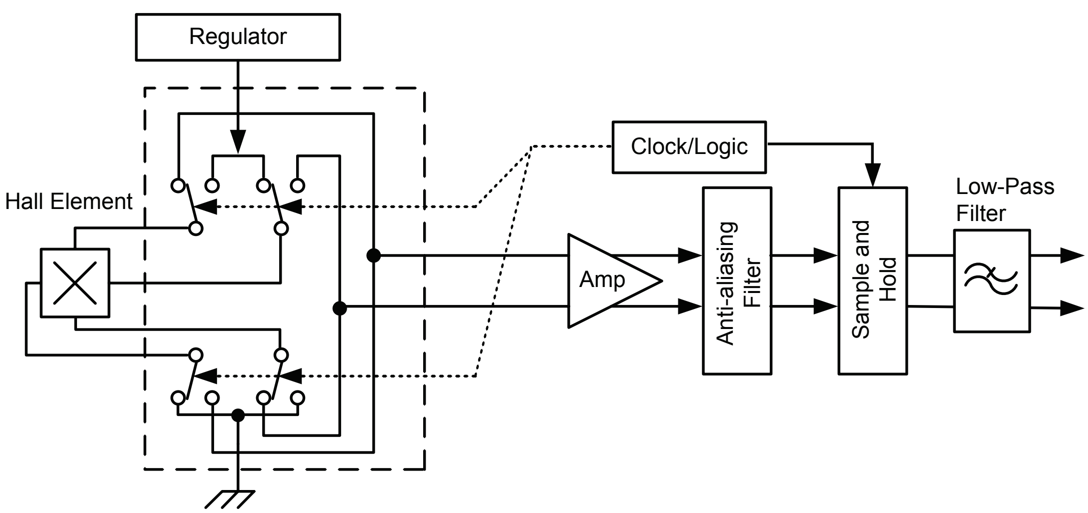
Chopper Stabilization Technique

Chopper Stabilization is an innovative circuit technique that is used to minimize the offset voltage of a Hall element and an associated on-chip amplifier. The technique nearly eliminates Hall IC output drift induced by temperature or package stress effects.

This offset reduction technique is based on a signal modulation-demodulation process. Modulation is used to separate the undesired DC offset signal from the magnetically induced signal in the frequency domain. Then, using a low-pass filter, the modulated DC offset is suppressed while the magnetically induced signal passes through the filter. The anti-aliasing filter prevents aliasing from happening in applications with high frequency signal components which are beyond the user's frequency range of interest.

As a result of this chopper stabilization approach, the output voltage from the Hall IC is desensitized to the effects of temperature and mechanical stress. This technique produces devices that have an extremely stable Electrical Offset Voltage, are immune to thermal stress, and have precise recoverability after temperature cycling.

This technique is made possible through the use of a BiCMOS process that allows the use of low-offset and low-noise amplifiers in combination with high-density logic integration and sample and hold circuits.

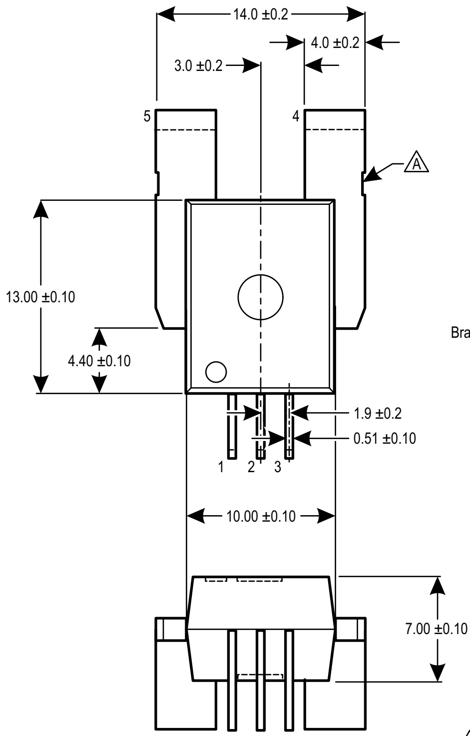


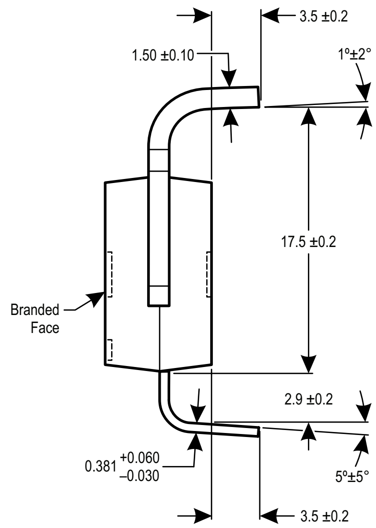
Concept of Chopper Stabilization Technique

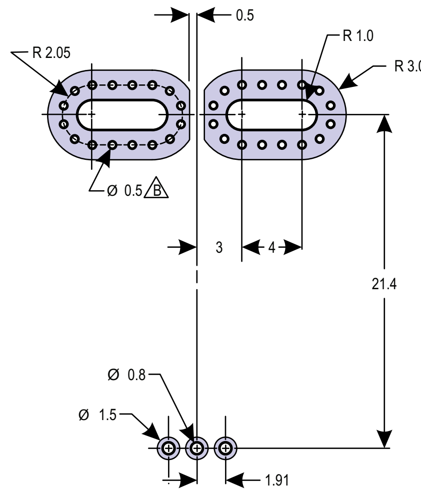
PACKAGE OUTLINE DRAWINGS

For Reference Only – Not for Tooling Use

(Reference DWG-9111 & DWG-9110)
Dimensions in millimeters – NOT TO SCALE
Dimensions exclusive of mold flash, gate burs, and dambar protrusions
Exact case and lead configuration at supplier discretion within limits shown

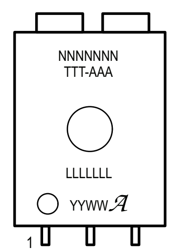






B PCB Layout Reference View

A Dambar removal intrusion
B Perimeter through-holes recommended
C Branding scale and appearance at supplier discretion

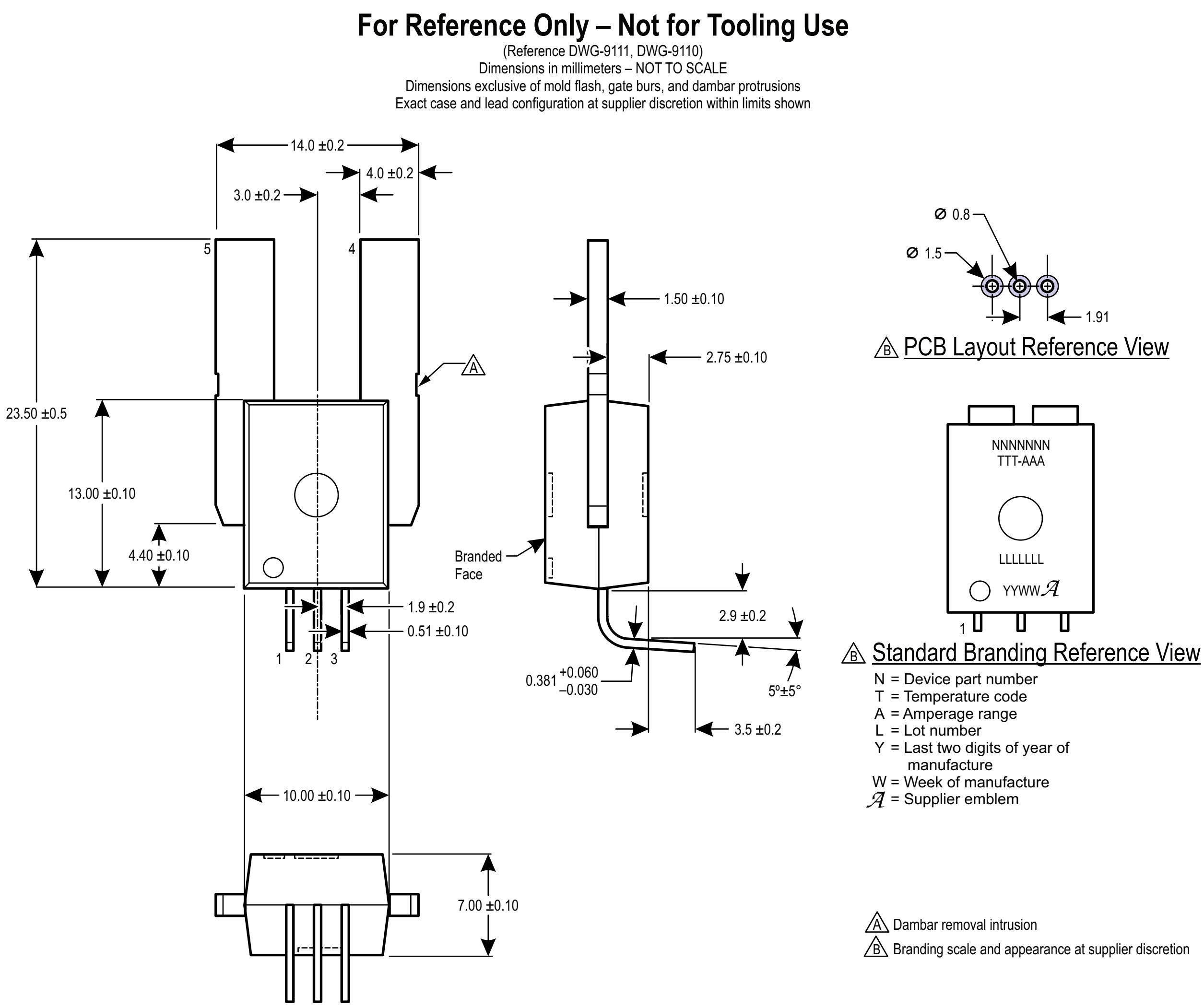


C Standard Branding Reference View

N = Device part number
T = Temperature code
A = Amperage range
L = Lot number
Y = Last two digits of year of manufacture
W = Week of manufacture
𝒜 = Supplier emblem

Creepage distance, current terminals to signal pins: 7.25 mm
Clearance distance, current terminals to signal pins: 7.25 mm
Package mass: 4.63 g typical

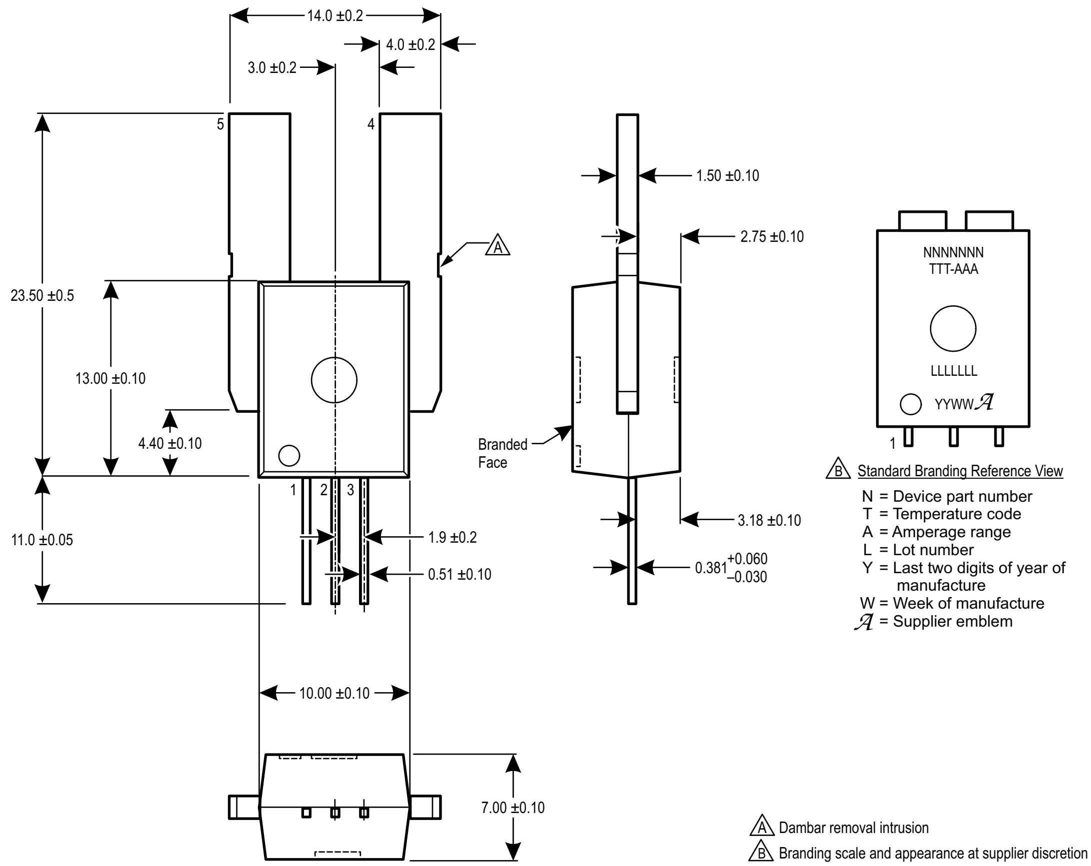
Package CB, 5-pin Package, Leadform PFF



Package CB, 5-pin Package, Leadform PSF

For Reference Only – Not for Tooling Use

(Reference DWG-9111, DWG-9110)
Dimensions in millimeters – NOT TO SCALE
Dimensions exclusive of mold flash, gate burs, and dambar protrusions
Exact case and lead configuration at supplier discretion within limits shown



Creepage distance, current terminals to signal pins: 7.25 mm
Clearance distance, current terminals to signal pins: 7.25 mm
Package mass: 4.63 g typical

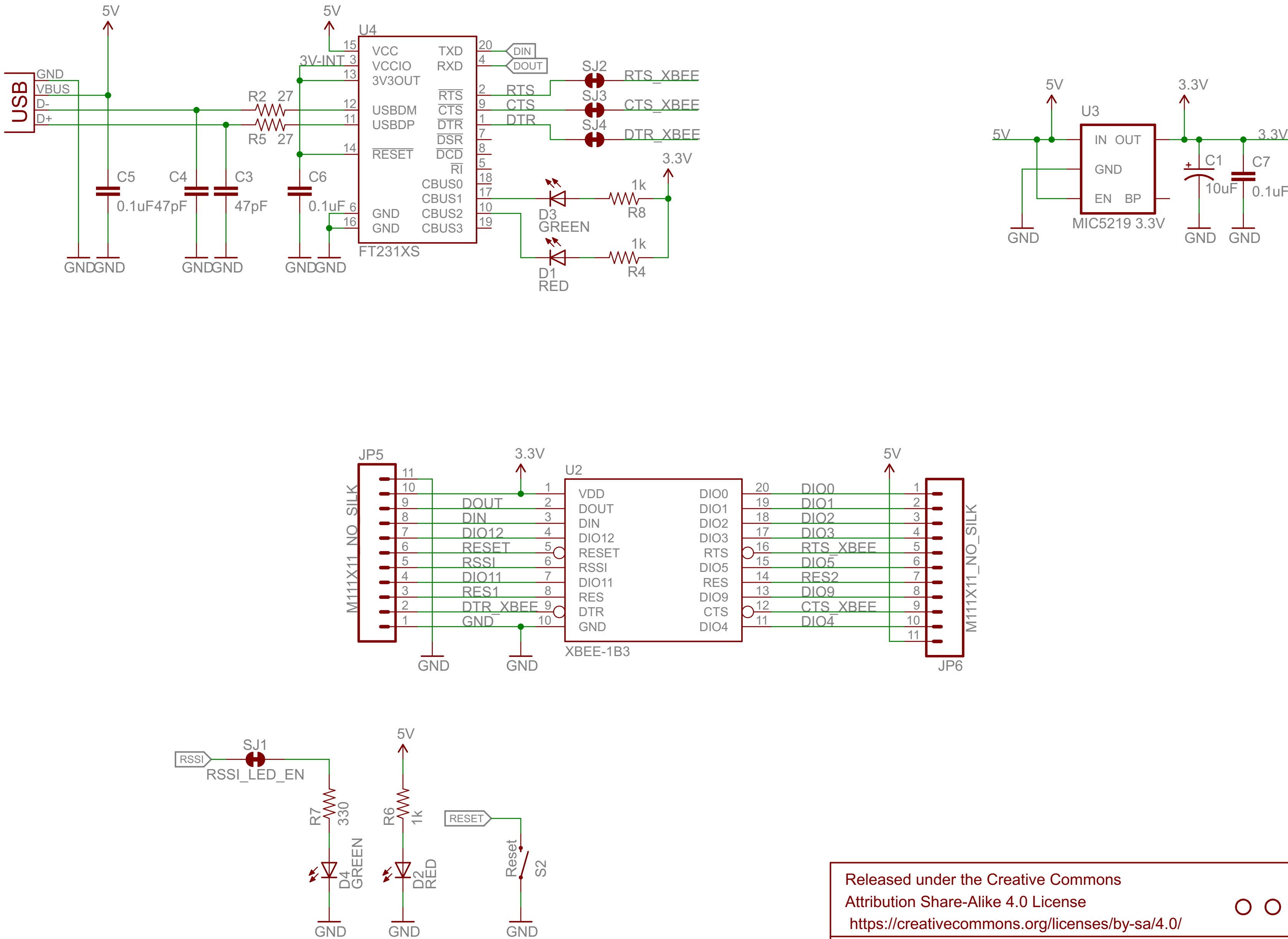
Package CB, 5-pin Package, Leadform PSS

Revision History

Number	Date	Description
8	January 17, 2014	Update features list and product offering.
9	April 7, 2015	Updated TUV certification and reformatted document.
10	November 17, 2016	Updated PCB Layout Reference View in Package Outline Drawing on page 19.
11	June 5, 2017	Updated product status

For the latest version of this document, visit our website:

www.allegromicro.com



TITLE: XBee-Explorer

SFE

Design by: Nathan Seidle, Jim Lindblom

REV: v21

Date: 6/3/2014 9:05:46 AM

Sheet: 1/1

open hardware